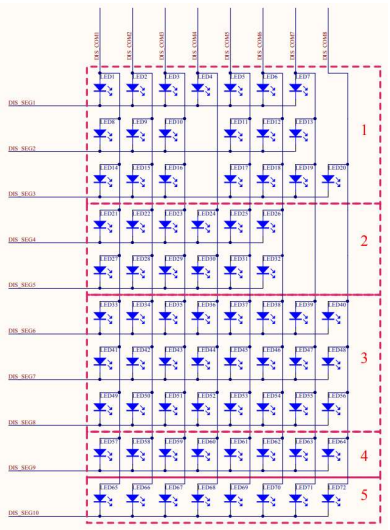
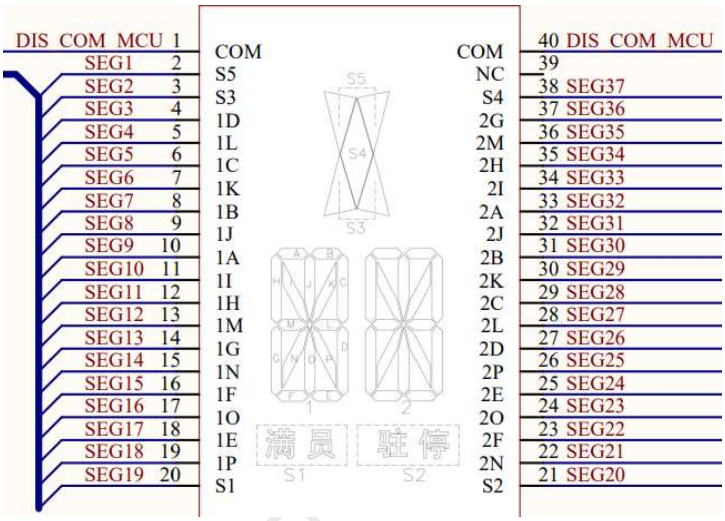
本文档仅介绍 ZK 公司通用的 LED、LCD 段码显示类产品的显示部分软件一般设计思路

一、确认硬件电路设计的 COM 口与 SEG 口数量





二、根据硬件电路灵活抽象出显示 RAM 缓冲区

如左上图 LED 段码 COM 端有 8 个口，SEG 端有 10 个段，此时可抽象出如下显示缓冲区

unsigned short gLedDispRamBuf[8]或 unsigned char gLedDispRamBuf[8][10]

两个缓冲区的差别在于前者是以 bit 单位来映射硬件电路上的一个 SEG 所对应的所有灯状态。后者则是以 byte 单位映射。

此时缓冲区中的 bit 值或是 byte 值状态即代表对应硬件电路上的 LED 灯或是 LCD 的 SEG 状态。

如右上图 LCD 段码 COM 端有 1 个口，SEG 端有 37 个 SEG。此时可抽象出如下显示缓冲区

unsigned char gLcdDispRamBuf[37];

之后根据不同的应用需求对 RAM 的数据进行操作，即可在硬件上呈现出对应的显示效果。

三、设定相应的定时器对显示缓冲区的数据进行扫描，控制对应的硬件 I/O 接口

扫描思路：

1、 选中对应的显示缓冲区映射的 COM 数据，控制对应的硬件 COM 口状态

2、 根据选中的 COM 数据，判断其对应的 bit 或是 byte 状态，控制对应的硬件 SEG 口状态

示例：

```
void timer_irq(void)
{
    HardWareComIO[gComIndex-1] = 0;释放上一个硬件COM选中状态
    HardWareComIO[gComIndex] = 1;选中当前需要的硬件COM
    for(i = 0; i < 10; i++){
        if(gLedDispRamBuf[gComIndex]&(1<<i)){
            HardWareSegIO[i]硬件SEG点亮
        }else{
            HardWareSegIO[i]硬件SEG熄灭
        }
    }
    gComIndex++;准备下一个即将选中的COM口
    ....;防止相应显示缓冲区溢出等相关逻辑处理
}
```