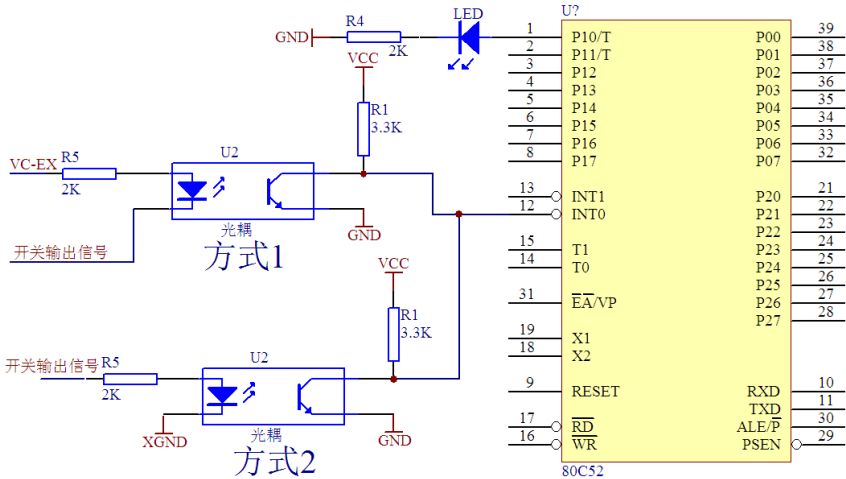


4.6 设 MCS-51 单片机 AT89C51 外部中断 0 连接到一接近开关，当接近开关检测到某一个物体时，输出高电平，否则输出低电平，且当接近开关输出高电平时单片机 P1.0 输出高电平，否则 P1.0 输出低电平。要求采用中断方式实现 P1.0 控制，绘制电路原理图并分别用汇编和 C 语言编写完整程序。

解答：电路图如下



电路图中，外部接近开关通过光耦实现单片机工作电源与外部传感器（接近开关电源）隔离，由于 MCS-51 单片机外部中断要么低电平触发，要么下降沿触发，因此需要对电平跳变逻辑转换，因此在实际使用中可以根据实际信号输出特性选择方式 1 或方式 2 的连接电路，其中方式 1 的逻辑为：开关输出高电平（逻辑 1 时），INT0 的电平为高电平，当开关输出低电平时，INT0 的电平为低电平；方式 2 的逻辑为：开关输出高电平（逻辑 1 时），INT0 的电平为低电平，当开关输出低电平时，INT0 的电平为高电平。

本题目中，条件是检测的物体，输出高电平，这一条件代表检测到物体时，CPU 应该发生中断；反之，没有检测到输出低电平时，不应产生中断，因此传感器信号到 CPU 的光耦隔离连接采用方式 2 合理；此外，接近开关检测到物体后，接近开关输出的高电平信号会保持一定时间，显然不能让 CPU 持续产生中断，因此采用下降沿触发中断合理。为此，

参考程序如下：

```
#include<reg51.h>
sbit LED=P1^0;
sbit P32=P3^2;
bit ExIrqFlag=0; //用于标识是否发生外部中断
//外部中断 0 服务程序
void EIRQ0(void) interrupt 0
{
    LED=1; //点亮发光二极管
    ExIrqFlag=1; //发生了外部中断
}
void main(void)
{
    unsigned char i;
    P3=0xff; //端口初始化，为外部中断信号输入作准备
    ExIrqFlag=0; //发生了外部中断
    IT0=1; //设置外部中断 0 为下降沿触发方式
    EX0=1; //外部中断 0 允许
    EA=1;
    LED=0; //LED 初始状态为点亮
    LED=0; //关闭发光二极管
```

---

```
while(1) //如果发生了外部中断，LED 被点亮了，则需要判断开关合适输出低电平
{
    if (ExIrqFlag==1)
    {
        if (P32==0)
        {
            for(i=0;i<50;i++); //此处延时是为了确保接近开关已经没有检测到物体
            if (P32==0)
            {
                LED=0; //关闭发光二极管
                ExIrqFlag=0; //清除外部中断标志
            }
        }
    }
}
```

程序中需要注意的是，退出外部中断后，接近开关不一定是低电平，因此不建议在中断服务程序中进行 P1.0 的低电平输出控制。