**任务 ：1用中断控制一键一灯**

**2参考课件中实验四的三键八灯代码用中断控制**

**补充知识：**

**1.什么是中断**

中断是一种异常情况，原本我正在看书，突然间有一个人敲我房间的门。导致我不得不放下手中的书，去开个门，看看是谁？找我有什么事？这里看书就被敲门声中断了。那么中断的行为是什么呢？我去开门，和敲门者聊聊天就是中断后的处理中断的行为。

我们把打断程序正常运行的过程称为**中断**。

把请求中断的外设称为**中断源**。

把处理中断的函数称之为**中断服务程序**。

并且针对于任何一个中断，都有一个唯一的**中断向量号**与之对应。

**2.为什么有中断这样异步处理的机制**

我们举一个最简单的例子就知道了。还是我在房间里面看书，如果不使用中断机制就好像是有人说他今天会来找我，但是他不会敲门，我要知道他来了没来，我今天就必须每隔个几分钟走到门口开个门，看看人来了没有。是不是很烦，其实我们之前写的按键扫描函数就是这样实现的，while(1)循环一直在检测有没有按键按下，这样是很耗费资源的。

如果说你一分钟看一次还好，如果说一个小时看一次，万一你刚刚打开门看完，人就来了，等到下一个小时，是不是人家就得在等59分钟，对于代码来说，这就是为什么按键的按键处理函数不能很好的实时被响应。可能你按下的时候，并没有在扫描。

**3.51单片机有哪些中断源**

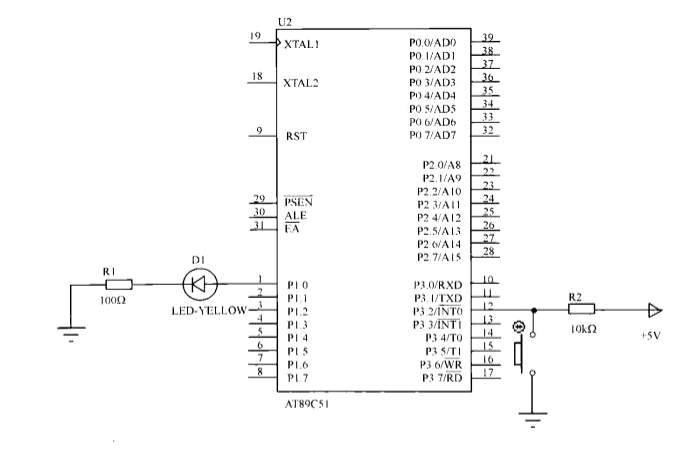
51单片机有5个中断源：

|  |  |  |
| --- | --- | --- |
| 中断源 | 中断向量号 | 说明 |
| 外部中断引脚/INT0 | 0 | 通过外部引脚P3.2口触发中断 |
| 定时器0溢出中断 | 1 | 内部定时器T0溢出触发中断 |
| 外部中断引脚/INT1 | 2 | 通过外部引脚P3.3口触发中断 |
| 定时器1溢出中断 | 3 | 内部定时器T1溢出触发中断 |
| 串口收发中断 | 4 | 串口接收/发送完成数据触发中断 |

所以为了节约CPU的资源，我们会用中断方式替代查询方式。并且同时提高了代码的实时性。

**4.设计一个最简单的外部中断触发电路**

上面说了通过外部引脚P3.2口可以触发外部中断0，是因为P3.2就复用了/INT0(外部中断0)。



一般就是直接一个按键接在/INT0口，按键按下会产生下降沿。按键持续按下会维持低电平。

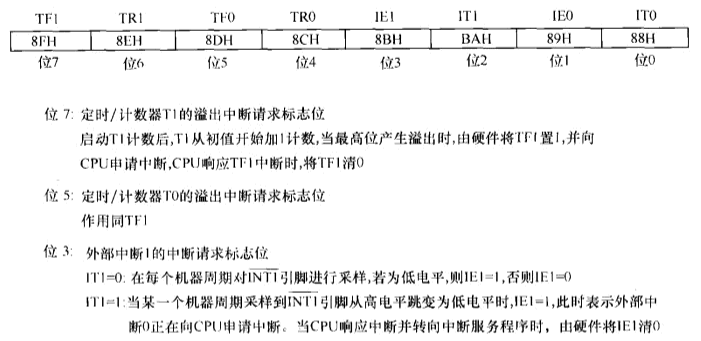
如果是多个按键公用/INT0，则会用与门连接。(具体实例请看本章实验)。

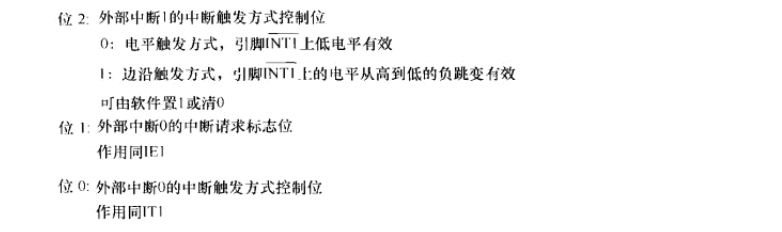
**5.外部中断的触发方式**

外部中断有两种触发方式：低电平触发和下降沿触发。

可以通过配置**TCON寄存器**设置外部中断的触发方式。请看IT0和IT1位。

**中断请求标志寄存器TCON**





**6.开启外部中断**

现在我们再回过头来看一下51单片机的P3口，我们不难发现51的P3口都是复用口，图中P3都标有第二功能。

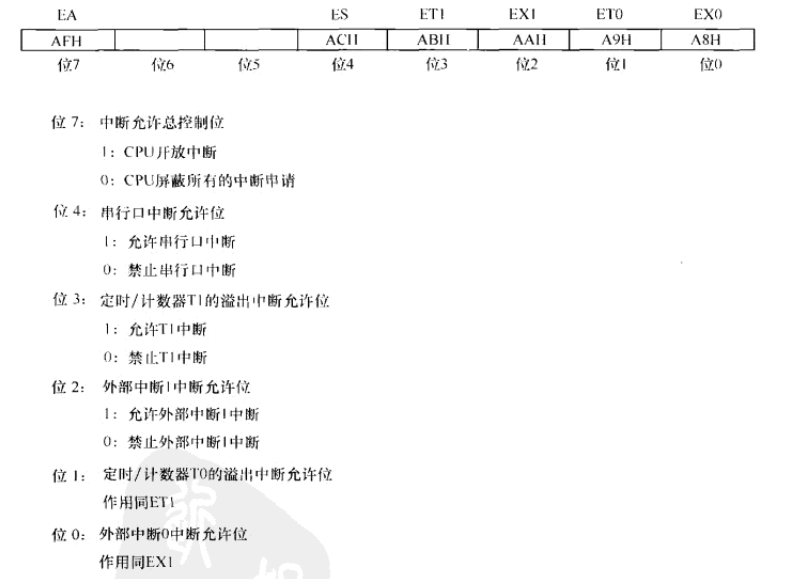
那么什么是复用口？其实就是一个I/O口复用了多个功能。举个例子，比如说P3.2既可以作为通用的I/O口也可以作为外部中断的触发引脚。

然后问题又来了，既然一个I/O口有多个功能，怎么知道它现在是使用哪一个功能。

对于P3口，默认的功能都是通用的I/O功能，需要使用它们的第二功能需要配置相关的寄存器。

对于外部中断触发引脚P3.2，在使用外部中断之前，需要先开启外部中断和设置好外部中断的触发方式。

**IE中断允许寄存器**



这里如果我们使用外部中断0的话，需要将IE的位0设置为1，同时也要设置IE的位7也设置为1。

**7.编写中断服务程序**

之前已经设计好了外部中断的电路连接，配置好了外部中断的触发方式，并且也开启了外部中断的中断允许位。

最后就只差一个，中断触发后该做些什么。这就是最后一步编写中断服务程序了。

前面说了51单片机有5个中断源，每一个中断源都对应有中断向量号。中断服务函数的编写就要用到中断向量号。

以前我们定义一个函数是这样定义的：

void function(int t)

{

}

中断服务函数需要在定义函数时指定中断向量号，来表征是哪一个中断的服务函数。

void function(int t) **interrupt 0**

{

}

通过interrupt关键字(51扩展的关键字)指定当前函数是外部中断0的中断服务函数。

**8.总结**

1) 设计中断触发电路，通过按键触发。

2) 设置TCON寄存器，设置外部中断0触发方式为下降沿触发。

3) 设置IE寄存器，开启外部中断0和总中断。

4) 编写外部中断服务函数。

按照上述操作，原本程序在中规中矩的运行主函数的程序。主函数并不需要主动的调用中断服务程序，但是一旦按键按下的一瞬间，就会触发外部中断，CPU立马停下手中的活(主函数的代码)，转到中断服务函数中运行中断服务函数，执行完了中断服务函数，再回到主函数中继续执行。

## 【元件清单】

|  |  |
| --- | --- |
| 元器件名称 | 说明 |
| AT89C51 | 主控芯片 |
| CHIPRES100R | 100欧电阻 |
| CHIPRES10K | 10K欧电阻 |
| LED-YELLOW | 黄色LED灯 |
| BUTTON | 按键 |
| 74LS04 | 非门 |
| 4068 | 8输入非与门 |