第5章

单片机中断系统设计



【学习指南】

通过本章的学习，了解单片机中断的基本原理，理解中断优先级和中断嵌套，学会设置中断优先级；掌握中断允许的条件，掌握中断标志的触发条件和清除要求，会编写中断服务程序。中断系统是单片机的一个重要功能，它保证一些级别较高或比较重要的事件能够避免等待，立刻执行，对于检测或控制实时性要求较高的场合非常有用。本章的难点在于无法掌握中断发生具体时刻或在哪行程序上中断，学习中主要理解中断工作原理，通过原理判断中断执行时机，从而熟练掌握中断系统。

5.1 中断系统概述

中断是指 CPU 在执行程序的过程中，有某随机事件发生，CPU 暂停正在执行的程序，转去执行处理该事件，处理完该事件后，又返回被中止的程序处继续执行。

在中断系统中，通常将 CPU 正常运行的程序称为主程序，把引起中断的设备或事件称为中断源。由中断源向 CPU 所发出的请求中断的信号称为中断请求信号。CPU 接受中断申请，终止当前程序，而转去执行中断服务程序称为中断响应。服务于中断事件的程序称为中断服务程序，也称为中断处理程序。当前程序被中断的地方称为断点，执行完中断服务程序后返回源程序断点处称为中断返回。

很多读者对中断很困惑，不能理解其含义，也不知道如何应用？举个例子，如果你正在网上看电影（运行主程序），突然热水壶响了（中断产生），你不得不去冲水（在主程序中断那一刻不再运行下一程序语句），你只得先把电影暂停（单片机自动操作），去冲水后（运行中断服务程序）再继续看电影，这就涉及许多中断概念；按电影暂停键，有个专有名词叫断点；等冲水结束了，你又继续看电影，等于又重新从断点继续。

中断不同于一般的查询方法，查询法查询开水状态会消耗主人大量精力，而热水壶会发出响声提醒主人水开，即热水壶（中断源）主动推送水开消息，主人收到消息后会执

行(中断响应),可以大大提高 CPU 效率。中断执行事件不会占用 CPU 大量时间,并且提高了响应事件效率,在一些工业领域的数据采集、监控等方面非常有用。中断流程如图 5-1 所示。

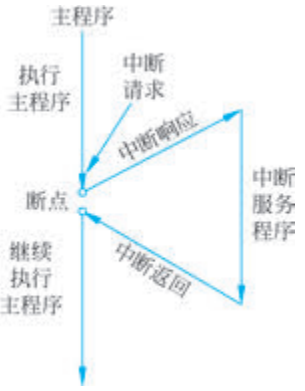


图 5-1　中断流程

5.2　中断优先级含义

一个单片机应用系统,特别是单片机实时测控系统,往往有多个中断源,各中断源所要求的处理具有不同的轻重、缓急程度。与人处理问题的思路一样,希望重要紧急的事件先处理。如果当前正在处理某个事件,有更重要、更紧急的事件到来,就应当暂停当前事件的处理,转去处理新事件,这就是中断系统优先级控制所要解决的问题。中断优先级的控制可以形成中断嵌套,见图 5-2。

所谓中断优先级,就是当有多个中断时优先执行哪一个。当中断正在进行时,优先级高的中断可以打断现有的中断,优先执行。如果你正在看电影(主程序),电话铃响了,你去接电话(响应一个中断),这时水又开了,你暂停电话先冲水(前一个中断没有结束又响应了一个中断),冲完水再接电话,接完电话后你继续看电影,这种方式叫中断嵌套。

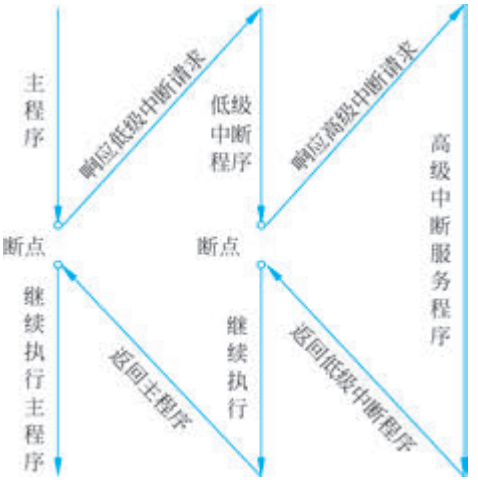


图 5-2　中断嵌套示意图

中断是如何发生的呢?中断处理过程可分为中断请求、中断响应、中断服务和中断返回。中断源发出中断申请,但要执行(响应)中断还要 CPU 许可,也叫中断允许条件,以图 5-3 为例说明。

MCS-51 单片机有 5 个中断源,其中 2 个外部中断、2 个定时器/计数器中断和 1 个串行口中断,下面分别介绍。

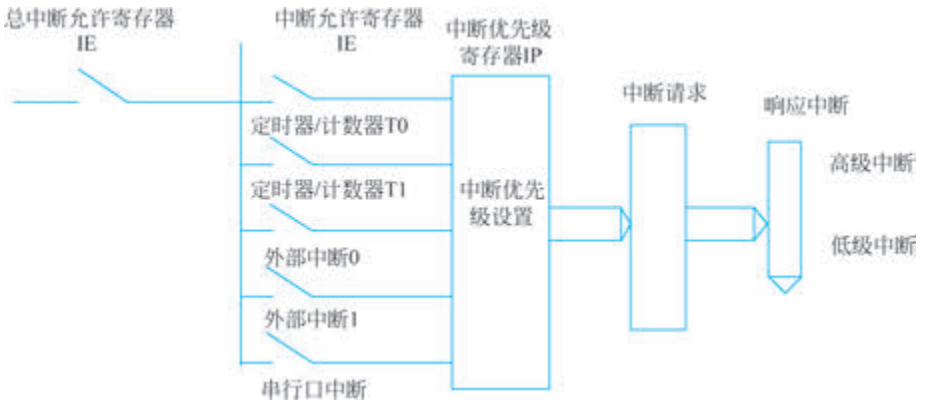


图 5-3 中断系统控制原理

(1) 外部中断源。外部中断是由外部请求信号引起的,共有 2 个中断源: $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$。$\overline{\text{INT0}}$: 外部中断 0,由 P3.2 端口线引入,低电平或下降沿触发。所谓低电平触发,是指只要是低电平,中断一直进行。所谓下降沿触发,即当端口由高电平变成低电平时,中断才能发生。$\overline{\text{INT1}}$: 外部触中断 1,由 P3.3 端口线引入,低电平或下降沿触发。

(2) 定时器/计数器中断源。单片机内有 2 个定时器/计数器 T0、T1,通过一种计数结构,实现定时/计数功能,当计数值发生溢出时,表明已经达到预期定时时间或计数值,定时器/计数器的中断请求标志位 TF0 或 TF1 被置 1,也就向 CPU 发出了中断请求的申请。

(3) 串行口中断源。当串行口接收或发送完一组数据时,便产生一个中断请求,特殊功能寄存器 SCON 中的 RI 或 TI 被置 1。TI/RI: 串行通信中断,完成一帧字符发送/接收后触发。

5.3 中断源及中断请求

5.3.1 中断源的中断入口地址

以 MCS-51 单片机为例,每个中断源都有对应的固定不变的中断入口地址,哪个中断源中断,CPU 就自动跳转到相应的中断入口地址执行程序。中断源及中断号见表 5-1。

表 5-1 中断源及中断号

中 断 源	C51 中断号
外部中断源 $\overline{\text{INT0}}$	0
定时器/计数器 T0 溢出中断	1
外部中断源 $\overline{\text{INT1}}$	2
定时器/计数器 T1 溢出中断	3
串行口中断源	4

5.3.2　中断请求

要实现中断，首先中断源要提出中断请求，单片机响应中断请求的过程是寄存器TCON和SCON相关状态位（中断请求标志位）置1的过程。CPU响应中断后，中断请求标志位要由硬件或软件清零。

定时器控制寄存器（TCON）用于保存外部中断请求，以及作为定时器/计数器的计数溢出标记；既可以对其整个字节寻址，又可以对其位寻址。寄存器地址88H，位地址8FH～88H，各位内容如表5-2所示。

表5-2　TCON的结构、位名称及其功能

TCON	D7	D6	D5	D4	D3	D2	D1	D0
位名称	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
功能	T1中断申请标志位	T1启动位	T0中断申请标志位	T0启动位	$\overline{\text{INT1}}$中断申请标志位	$\overline{\text{INT1}}$触发方式	$\overline{\text{INT0}}$中断申请标志位	$\overline{\text{INT0}}$触发方式

TCON寄存器既有定时器/计数器的控制功能，又有中断控制功能，其中与中断有关的控制位共6位：IE0和IE1、IT0和IT1以及TF0和TF1。

TR0：定时器/计数器T0运行启停控制位（由用户通过软件设置）。

TR0＝0：定时器/计数器T0停止运行；

TR0＝1：定时器/计数器T0启动运行。

TR1：定时器/计数器T1运行启停控制位（由用户通过软件设置）。

TR1＝0：定时器/计数器1停止运行；

TR1＝1：定时器/计数器1启动运行。

TF0：定时器/计数器T0溢出中断申请标志位（由硬件自动置位）。

TF0＝0时定时器/计数器T0未溢出；

TF0＝1时定时器/计数器T0溢出（由全“1”变成全“0”），此时由硬件自动置位，申请中断，中断被CPU响应后由硬件自动清零，使用定时器/计数器但没有采用中断模式，需要手工清零。

TF1：定时器/计数器T1溢出中断申请标志位（由硬件自动置位）。

TF1＝0时定时器/计数器T1未溢出；

TF1＝1时定时器/计数器T1溢出（由全“1”变成全“0”），此时由硬件自动置位，申请中断，中断被CPU响应后由硬件自动清零，使用定时器/计数器但没有采用中断模式，需要手工清零。

IE0：外部中断0申请标志位（由硬件自动置位，中断响应后转向中断服务程序时，由硬件自动清零）。

IE0＝0时没有外部中断0申请；

IE0＝1时有外部中断0申请。

IE1：外部中断1申请标志位（由硬件自动置位，中断响应后转向中断服务程序时，由硬件自动清零）。

IE0＝0时没有外部中断1申请；

IE1=1 时有外部中断 1 申请。

IT0：外部中断 0 请求的触发方式控制位(可由用户通过软件设置)。

IT0=0 时在 $\overline{\text{INT0}}$ 端申请中断的信号低电平有效；

IT0=1 时在 $\overline{\text{INT0}}$ 端申请中断的信号下降沿有效。

IT1：外部中断 1 请求的触发方式控制位(可由用户通过软件设置)。

IT1=0 时在 $\overline{\text{INT1}}$ 端申请中断的信号低电平有效；

IT1=1 时在 $\overline{\text{INT1}}$ 端申请中断的信号下降沿有效。

SCON 中的串行中断标志位，SCON 寄存器地址 98H，位地址 9FH～98H，各位内容如表 5-3 所示。

表 5-3 SCON 的结构、位名称及其功能

SCON	D7	D6	D5	D4	D3	D2	D1	D0
位名称	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
功能	—	—	—	—	—	—	串行口发送中断申请标志位	串行口接收中断申请标志位

其中，高 6 位用于串行口工作方式设置和串行口发送/接收控制，低 2 位 RI 和 TI 锁存串行口的接收中断和发送中断的申请标志位。由硬件自动置位，但必须由用户在程序中用软件清零。

TI=0 时没有串行口发送中断申请；TI=1 时有串行口发送中断申请。

RI=0 时没有串行口接收中断申请；RI=1 时有串行口接收中断申请。

提示：表 5-2 以及表 5-3 中的 TI、RI 等内容是中断应用的基础，要好好掌握，其标志位要记牢。

5.4 中断允许控制

中断源申请中断后，中断能否被响应，取决于 CPU 对中断源的开放或屏蔽状态，由内部的中断允许寄存器 IE 进行控制，IE 的地址是 A8H，位地址为 AFH～A8H，各位内容如表 5-4 所示。

表 5-4 IE 的结构、位名称及其功能

IE	D7	D6	D5	D4	D3	D2	D1	D0
位名称	EA	—	—	ES	ET1	EX1	ET0	EX0
功能	总开关	—	—	串行口	T1	$\overline{\text{INT1}}$	T0	$\overline{\text{INT0}}$

EA：总的中断允许控制位(总开关)，EA=0 时禁止全部中断；EA=1 时允许中断。

EX0：$\overline{\text{INT0}}$ 的中断允许控制位。EX0=0 时禁止 $\overline{\text{INT0}}$ 中断；EX0=1 时允许 $\overline{\text{INT0}}$ 中断。

EX1：$\overline{\text{INT1}}$ 的中断允许控制位。EX1=0 时禁止 $\overline{\text{INT1}}$ 中断；EX1=1 时允许 $\overline{\text{INT1}}$ 中断。

ET0：T0 的中断允许控制位。ET0＝0 时禁止 T0 中断；ET0＝1 时允许 T0 中断。

ET1：T1 的中断允许控制位。ET1＝0 时禁止 T1 中断；ET1＝1 时允许 T1 中断。

ES：串行口的中断允许控制位。ES＝0 时禁止串行口中断；ES ＝1 时允许串行口中断。

提示：表 5-4 是中断应用的基础，要好好掌握，其标志位建议记牢。

5.5　中断优先权管理

MCS-51 单片机有两个中断优先级，每个中断源均可通过软件设置为高优先级或低优先级中断，实现 2 级中断嵌套。特殊功能寄存器 IP 为中断优先级控制寄存器，其地址为 B8H，位地址为 BFH～B8H，各位内容如表 5-5 所示。

表 5-5　IP 的结构、位名称及其功能

IP	D7	D6	D5	D4	D3	D2	D1	D0
位名称	—	—	—	PS	PT1	PX1	PT0	PX0
功能	—	—	—	串行口	T1	$\overline{\text{INT1}}$	T0	$\overline{\text{INT0}}$

PX0：外部中断 0 中断优先级控制位。PX0＝1，外部中断 0 定义为高优先级中断；PX0＝0，定义为低优先级中断。

PT0：定时器/计数器 T0 中断优先级控制位。PT0＝1，定时器/计数器 T0 中断定义为高优先级中断；PT0＝0，定义为低优先级中断。

PX1：外部中断 1 中断优先级控制位。PX1＝1，外部中断 1 定义为高优先级中断；PX1＝0，定义为低优先级中断。

PT1：定时器/计数器 T1 中断优先级控制位。PT1＝1，定时器/计数器 T1 中断定义为高优先级中断；PT1＝0，定义为低优先级中断。

PS：串行口中断优先级控制位。PS＝1 时，串行口中断定义为高优先级中断；PS＝0 时，定义为低优先级中断。

当系统复位后，IP 的所有位被清零，所有中断源均被定义为低优级中断。IP 的各位都可用程序置位和复位，可用位操作指令或字节操作指令更新 IP 的内容，以改变各中断源的中断优先级。

MCS-51 单片机中断系统，遵循下列基本准则：

（1）低优先级中断可以被高优先级中断请求中断，高优先级中断不能被低优先级或同优先级中断请求中断。

（2）多个同级别中断源同时提出中断申请时，在默认状态下，响应顺序取决于内部规定的顺序，即自然优先级：外部中断 0、定时中断 0、外部中断 1、定时中断 1、串行中断，外部中断 0 优先级最高，串行中断优先级最低，如表 5-6 所示。

表 5-6 中断源优先级排列

中 断 源	同级自然优先级
外部中断 0	最高级
定时器/计数器 T0	↑
外部中断 1	
定时器/计数器 T1	
串行口中断	最低级

5.6 中断响应

5.6.1 中断响应条件

(1) IE 寄存器中的总中断允许位 EA=1。
(2) 该中断源发出中断请求,即该中断源对应的中断请求标志为“1”。
(3) 该中断源的中断允许位 EX0/EX1/ET0/ET1/ES=1,即该中断没有被屏蔽。
(4) 无同级或更高级中断正在被服务。
CPU 响应中断的过程如下:
(1) 将相应的优先级状态触发器置 1(以阻断后来的同级或低级的中断请求);
(2) 自动执行一条硬件跳转指令,即把程序计数器 PC 的内容压入堆栈保存;
(3) 跳到中断函数,执行中断服务程序。

5.6.2 中断现场保护和恢复

中断的现场保护主要是在中断时保存单片机存储单元中的数据和状态。中断的恢复是保存单片机在被中断前存储单元中的数据和状态。

保护现场就是在程序进入中断服务程序入口之前,将相关寄存器的内容、标志位状态等压入堆栈保存,避免在运行中断服务程序时破坏这些数据或状态,保证中断返回后,主程序能够正常运行。

中断程序中的现场保护,主要是对变量传递的控制,注意使用全局变量或局部变量实现保护现场的目的。

5.6.3 开关中断和中断标志位的清除

对于一个不允许在执行中断服务程序时被打扰的重要中断,可以在进入中断服务程序时关闭中断系统,在执行完中断服务程序后,再开放中断系统。由于 MCS-51 单片机不具有自动关闭中断的功能,因此进入服务子程序后,必须通过指令关闭中断。然后,在执行完中断服务程序返回断点之前再通过指令打开中断,允许响应别的中断请求。

在电平触发的外部中断请求响应后,硬件不能自动对中断请求标志位 IE0 或 IE1 清零。中断的撤除,需要撤除 $\overline{\text{INT0}}$ 或 $\overline{\text{INT1}}$ 引脚上的低电平才能有效。

对于串口中断,无论是发送中断产生,还是接收中断产生,硬件都不能自动清除 TI 和 RI 标志,需要手工清除。

其他中断被响应后,其中断请求标志都由硬件自动清除,无须采取其他措施。

表 5-7 对中断的相关知识点进行了总结，有助于增进对中断系统的了解。

表 5-7　相关中断内容总结

中断源	中断允许控制位	中断请求标志位	中断触发方式	优先级设置位	中断标志位清除方式
外部中断源 0	EX0＝1 允许 EX0＝0 禁止	IE0＝1 有请求 IE0＝0 无请求	IT0＝1 脉冲触发 IT0＝0 电平触发	PX0＝1 高级 PX0＝0 低级	跳沿触发的硬件自动清零 电平触发的低电平撤销
定时器/计数器 T0 溢出中断	ET0＝1 允许 ET0＝0 禁止	TF0＝1 有请求 TF0＝0 无请求	TR0＝1 开始启动 TR0＝0 停止工作	PT0＝1 高级 PT0＝0 低级	硬件自动清零
外部中断源 1	EX1＝1 允许 EX1＝0 禁止	IE1＝1 有请求 IE1＝0 无请求	IT1＝1 脉冲触发 IT1＝0 电平触发	PX1＝1 高级 PX1＝0 低级	跳沿触发的硬件自动清零 电平触发的低电平撤销
定时器/计数器 T1 溢出中断	ET1＝1 允许 ET1＝0 禁止	TF1＝1 有请求 TF1＝0 无请求	TR1＝1 开始启动 TR1＝0 停止工作	PT1＝1 高级 PT1＝0 低级	硬件自动清零
串行口中断源	ES＝1 允许 ES＝0 禁止	TI＝1 有发送请求 TI＝0 无发送请求 RI＝1 有接收请求 RI＝0 无接收请求		PS＝1 高级 PS＝0 低级	软件清零

5.7　C51 的中断函数格式

C51 的中断服务程序是一种特殊的程序，其定义如下：

```
void  函数名(void)  interrupt  n  using  m
      { 函数体语句; }
```

interrupt 和 using 是为编写 C51 中断服务程序而引入的关键字，interrupt 表示该函数是一个中断服务程序，其后的 n 是指该中断服务程序对应于哪个中断源，见表 5-1；using 用于指定该中断服务程序要使用的工作寄存器组编号，m 的取值为 0～3。

若不使用关键字 using，则编译系统会自动给该服务函数分配寄存器组，并将当前工作寄存器组的 8 个寄存器压入堆栈。

满足中断条件后，中断服务程序会自动执行，用户程序中不允许任何程序调用中断服务程序。

例 1　请设置下列中断源的优先级，满足以下条件：从左向右优先级降低，定时器/计数器 T0、外部中断 1、串行口中断、外部中断 0、定时器/计数器 T1，并进行初始化编程。

解题过程：先按默认优先级顺序排列中断源如下：

外部中断 0、定时器/计数器 T0、外部中断 1、定时器/计数器 T1、串行口中断

0　　1　　1　　0　　1

用位操作指令编写程序段如下：

```
PT0 = 1   ;定时器/计数器 0 中断优先级置 1
PX1 = 1   ;外部中断 1 中断优先级置 1
PS = 1    ;串行口中断优先级置 1
PX0 = 0   ;外部中断 0 中断优先级置 0
PT1 = 0   ;定时器/计数器 1 中断优先级置 0
```

或用字节操作语句编写为 IP＝0x16

例 2　若允许外部中断 0 和外部中断 1 中断，禁止其他中断源的中断请求，请编写设置 IE 的相应程序段。

解题过程：用位操作指令编写程序段如下：

```
ES = 0    ;禁止串口中断
ET1 = 0   ;禁止定时器/计数器 T1 中断
EX1 = 1   ;允许外部中断 1 中断
ET0 = 0   ;禁止定时器/计数器 T0 中断
EX0 = 1   ;允许外部中断 0 中断
EA = 1    ;CPU 开中断
```

或用字节操作语句编写为 IE＝0x85

例 3　见图 5-4，采用外部中断方法用按钮 SW1 控制指示灯 P1.0 的亮灭状态。

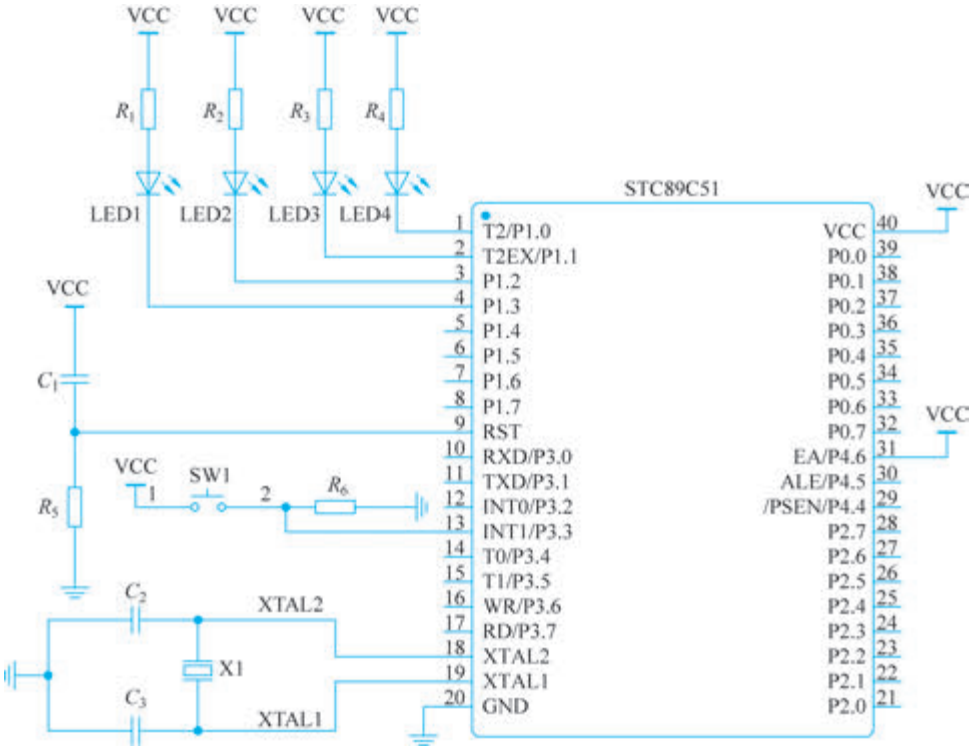


图 5-4　LED 控制电路

解题过程：

```
#include <reg51.h>              //寄存器声明头文件
sbit LED = P1^0;                //发光二极管控制引脚位定义
```

```
void INT1_srv(void) interrupt 2 using 1 //外部中断 0 处理程序
{
  LED = ! LED;                    //指示灯状态翻转
}
void main( )
{
  LED = 1;                        //关闭指示灯
  IT1 = 1;                        //外部中断 1 为边沿触发方式
  EA = 1;                         //中断允许
  EX1 = 1;                        //外部中断 1 允许
  while( 1 );
}
```

习题与思考

5-1　什么是中断？中断与查询程序及子程序有什么区别？

5-2　什么是中断优先级？MCS-51 单片机有几个优先级，顺序怎么排列？

5-3　中断响应的条件是什么？

5-4　外部触发中断的触发方式有哪几种，有什么区别？如何分别清除中断标志？

5-5　用外部中断 1 的中断方式控制 P1 口 1 盏发光管的亮暗，要求每按一次开关 K，灯由亮变暗或由暗变亮，见图 5-5，请编程实现。

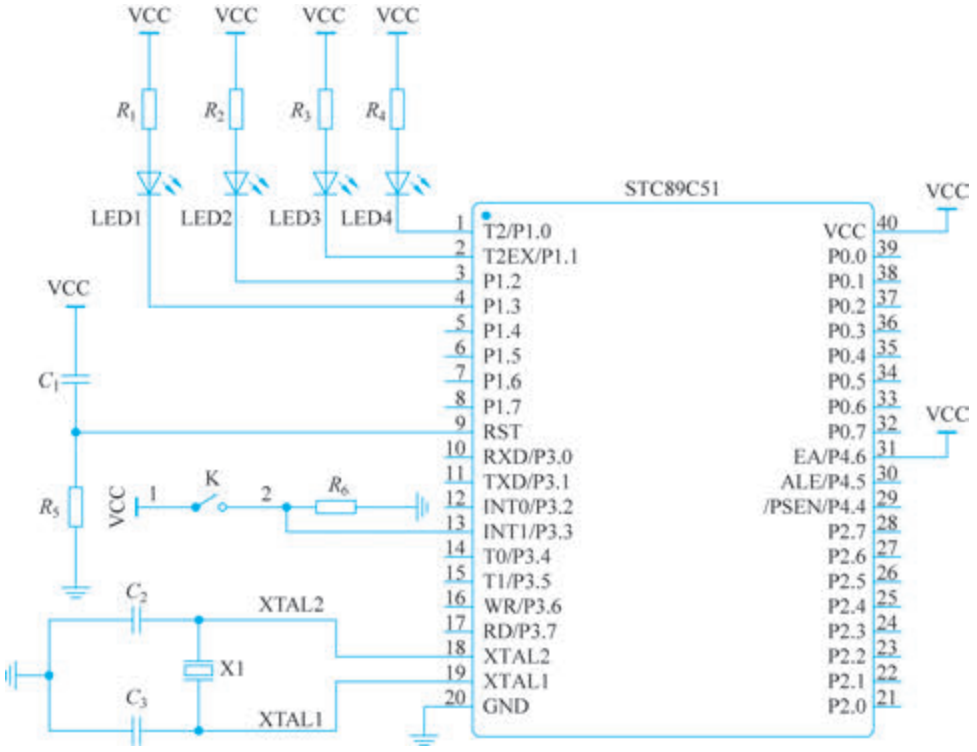


图 5-5　按键控制 LED 电路

5-6　简述单片机系统产生中断的完整过程。