

弟子教你学51单片机技术



从零到项目开发学51单片机 与C语言技术

基于DZ80C52开发板

QQ:2539606882

从零到项目开发学51单片机与C语言技术视频教程

弟子教你学51单片机技术



第九讲 单片机中断系统

基于DZ80C52开发板

QQ:2539606882

从零到项目开发学51单片机与C语言技术视频教程



中断

本讲任务:

了解中断概念，以INT外部中断为例详细讲解
中断程序的编写方法



引言

单片机中的中断系统是解决突发事件和多任务实时处理的最佳方法，熟练掌握中断技术的应用是一个合格电子工程师必备的能力。



为什么要引入中断原因

一个高速主机和一个低速外设连接时，效率极低，低速外设工作时无端占用大量CPU时间。

一个高速主机和多个低速外设连接时，高速主机无法进行多任务并行处理。



中断概念的引出

1、生活中的中断法例子。

2、中断的概念：

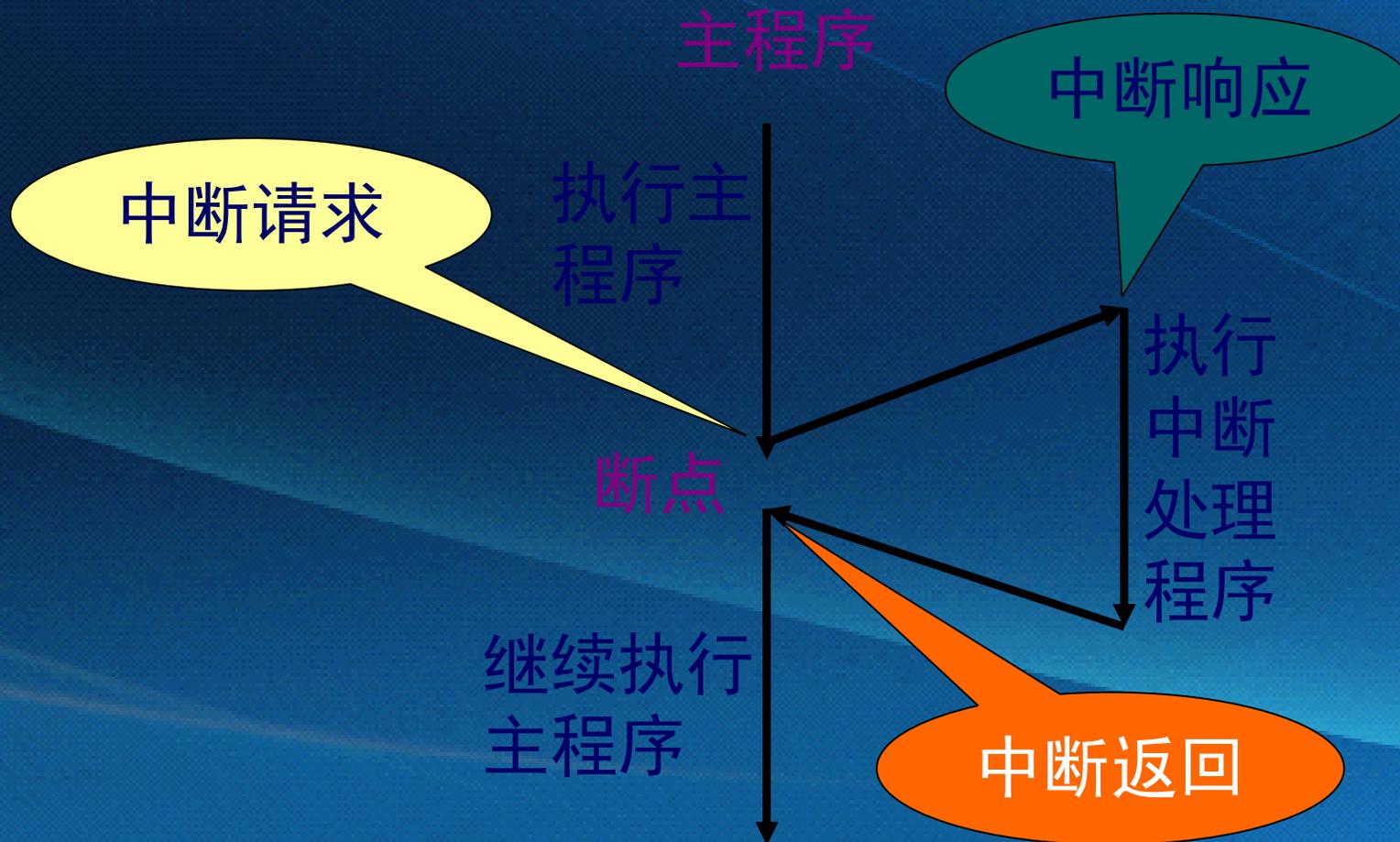
CPU在处理某一事件A时，发生了另一事件B的请求（**中断请求**）；

CPU暂时中断当前的工作，转去处理事件B（**中断响应和中断服务**）；

待CPU将事件B处理完毕后，再回到原来事件A被中断的地方继续处理事件A（**中断返回**），这一过程称为**中断**。



MCS-51单片机的中断系统结构



中断源：引起CPU中断的根源或原因。

中断源向CPU提出的中断请求。

中断发生的时刻是不能事先确定的。



引起CPU中断的根源，称为中断源。中断源向CPU提出的中断请求。CPU暂时中断原来的事务A，转去处理事件B。对事件B处理完毕后，再回到原来被中断的地方（即断点），称为中断返回。实现上述中断功能的部件称为中断系统（中断机构）。



中断的优点

早期的计算机系统是不包含中断系统的。

后来为了解决快速主机与慢速外设的数据传送问题，引入了中断系统，它的优点：

分时操作。**CPU**可以分时为多个外设服务，提高了计算机的利用率；

实时响应。**CPU**能够及时处理应用系统的随机事件，系统的实时性大大增强；

可靠性高。**CPU**具有处理设备故障及掉电等突发性事件能力，从而使系统可靠性提高。



中断传送方式特点

数据传送的双方平时各自做自己的工作，一旦甲方要求与乙方进行数据传送，就主动发出信号提出申请，乙方接到申请后若同意传送，安排好当前的工作，再响应与甲方发生数据传送。完事后，回去继续做打断前的工作。

- 中断功能强弱是计算机性能优劣的重要标志
 - ✓ 提高CPU效率
 - ✓ 解决速度矛盾
 - ✓ 实现并行工作
 - ✓ 应付突发事件.....

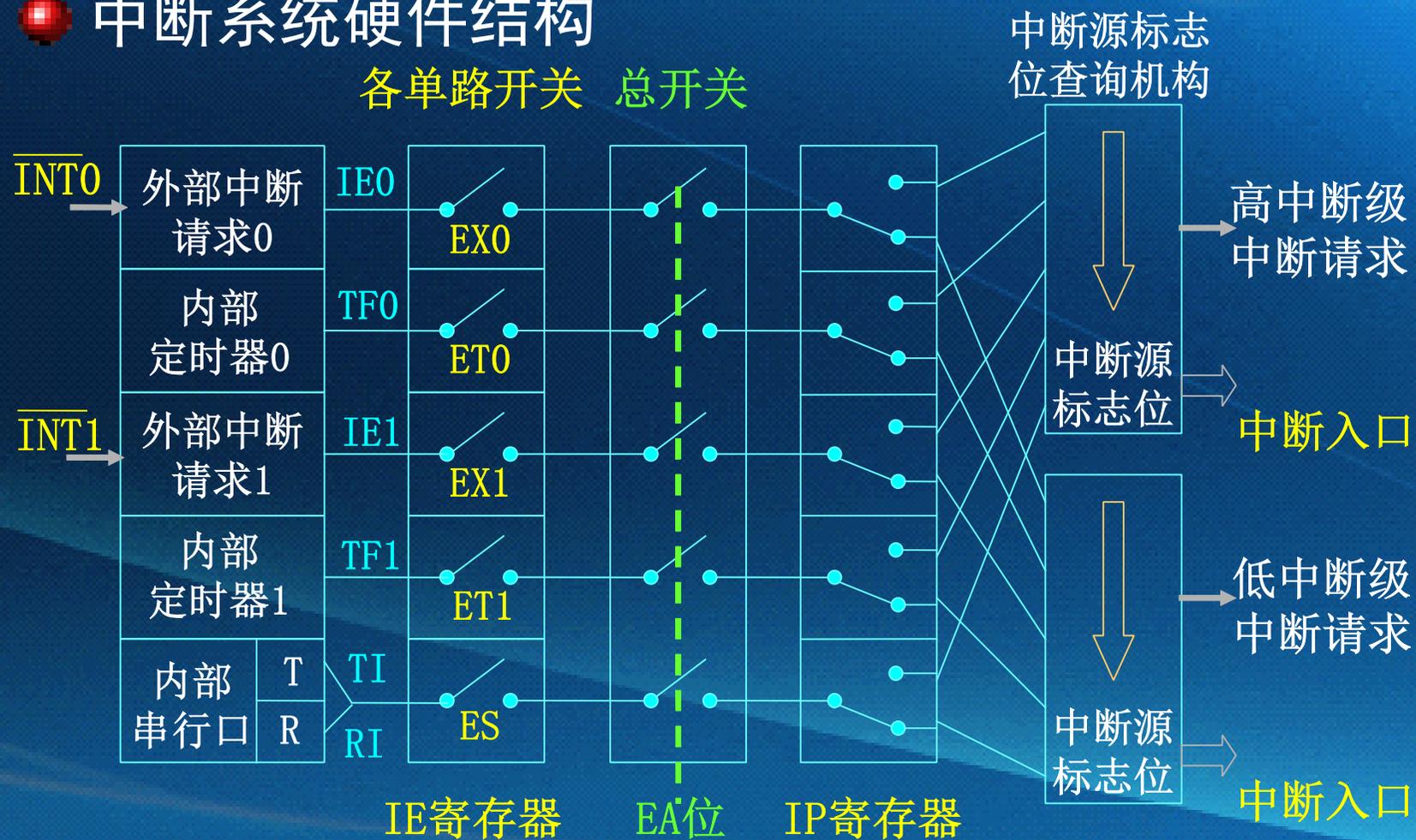


● 中断涉及的几个环节

- ①中断源 (前面提到的甲方)
- ②中断申请 (甲方发出信号提出申请)
- ③开放中断 (乙方同意传送)
- ④保护现场 (安排好当前的工作)
- ⑤中断服务 (响应乙方的要求)
- ⑥恢复现场 (完事后, 回去.....)
- ⑦中断返回 (继续做打断前的工作)



中断系统硬件结构



注:各中断允许控制位=0, 开关断开; =1, 开关接通



80C51中断系统的结构

80C51的中断系统有5个中断源（8052有6个），2个优先级，可实现二级中断嵌套。



1、(P3.2)可由IT0(TCON.0)选择其为低电平有效还是下降沿有效。当CPU检测到P3.2引脚上出现有效的中断信号时，中断标志IE0(TCON.1)置1，向CPU申请中断。



2、(P3.3)可由IT1(TCON.2)选择其为低电平有效还是下降沿有效。当CPU检测到P3.3引脚上出现有效的中断信号时，中断标志IE1(TCON.3)置1，向CPU申请中断。



3、TF0 (TCON.5)，片内定时/计数器T0溢出中断请求标志。当定时/计数器T0发生溢出时，置位TF0，并向CPU申请中断。



4、TF1 (TCON. 7)，片内定时/计数器T1溢出中断请求标志。当定时/计数器T1发生溢出时，置位TF1，并向CPU申请中断。



5、RI (SCON.0) 或TI (SCON.1)，串行口中断请求标志。当串行口接收完一帧串行数据时置位RI或当串行口发送完一帧串行数据时置位TI，向CPU申请中断。



SCON的中断标志

位	7	6	5	4	3	2	1	0	
字节地址: 98H							TI	RI	SCON

- RI (SCON.0)， 串行口接收中断标志位。当允许串行口接收数据时，每接收完一个串行帧，由硬件置位RI。注意，RI必须由软件清除。
- TI (SCON.1)， 串行口发送中断标志位。当CPU将一个发送数据写入串行口发送缓冲器时，就启动了发送过程。每发送完一个串行帧，由硬件置位TI。CPU响应中断时，不能自动清除TI，TI必须由软件清除。



80C51中断的控制

一、中断允许控制

CPU对中断系统所有中断以及某个中断源的开放和屏蔽是由中断允许寄存器**IE**控制的。

位	7	6	5	4	3	2	1	0	
字节地址: A8H	EA			ES	ET1	EX1	ET0	EX0	IE

- **EX0(IE.0)**, 外部中断**0**允许位;
- **ET0(IE.1)**, 定时/计数器**T0**中断允许位;
- **EX1(IE.2)**, 外部中断**0**允许位;
- **ET1(IE.3)**, 定时/计数器**T1**中断允许位;
- **ES (IE.4)**, 串行口中断允许位;
- **EA (IE.7)**, **CPU**中断允许 (总允许) 位。



中断优先级控制

80C51单片机有两个中断优先级，即可实现二级中断服务嵌套。每个中断源的中断优先级都是由中断优先级寄存器IP中的相应位的状态来规定的。

位 ₇	7 ₀	6 ₀	5 ₀	4 ₀	3 ₀	2 ₀	1 ₀	0 ₀	₀
字节地址: B8H ₀	₀	₀	PT2 ₀	PS ₀	PT1 ₀	PX1 ₀	PT0 ₀	PX0 ₀	IP ₀

- **PX0 (IP.0)**，外部中断0优先级设定位；
- **PT0 (IP.1)**，定时/计数器T0优先级设定位；
- **PX1 (IP.2)**，外部中断0优先级设定位；
- **PT1 (IP.3)**，定时/计数器T1优先级设定位；
- **PS (IP.4)**，串行口优先级设定位；
- **PT2 (IP.5)**，定时/计数器T2优先级设定位。



而80C52单片机有四个中断优先级，即可实现四级中断服务嵌套。每个中断源的中断优先级由中断优先级寄存器IP和IPH中的相应位的状态来规定的。

位	7	6	5	4	3	2	1	0	
字节地址: B7H			PT2	PS	PT1	PX1	PT0	PX0	IPH

- PX0 (IPH. 0)，外部中断0优先级设定位；
- PT0 (IPH. 1)，定时/计数器T0优先级设定位；
- PX1 (IPH. 2)，外部中断0优先级设定位；
- PT1 (IPH. 3)，定时/计数器T1优先级设定位；
- PS (IPH. 4)，串行口优先级设定位；
- PT2 (IPH. 5)，定时/计数器T2优先级设定位。



同一优先级中的中断申请不止一个时，则有中断优先权排队问题。同一优先级的中断优先权排队，由中断系统硬件确定的自然优先级形成，其排列如图所示：

各中断源响应优先级及中断服务程序入口表

中断源	中断标志	中断服务程序入口	优先级顺序
外部中断 0 ($\overline{\text{INT0}}$)	IE0	0003H	高
定时/计数器 0 (T0)	TF0	000BH	↓
外部中断 1 ($\overline{\text{INT1}}$)	IE1	0013H	↓
定时/计数器 1 (T1)	TF1	001BH	↓
串行口	RI 或 TI	0023H	低



- 设置52单片机的4个中断源,使他们的优顺序为T1, INT1, INT0, T0.

IPH = 0X08; PT1 = 1;

IP = 0X40; PX1 = 1;



80C51单片机的中断优先级有三条原则：

- CPU同时接收到几个中断时，首先响应优先级别最高的中断请求。
- 正在进行的中断过程不能被新的同级或低优先级的中断请求所中断。
- 正在进行的低优先级中断服务，能被高优先级中断请求所中断。

为了实现上述后两条原则，中断系统内部设有两个用户不能寻址的优先级状态触发器。其中一个置1，表示正在响应高优先级的中断，它将阻断后来所有的中断请求；另一个置1，表示正在响应低优先级中断，它将阻断后来所有的低优先级中断请求。



80C51单片机中断处理过程

中断响应条件和时间

中断响应条件

- 中断源有中断请求；
- 此中断源的中断允许位为1；
- CPU开中断（即EA=1）。

以上三条同时满足时，CPU才有可能响应中断。



如何进行中断编程

中断源:

8051的中断系统有5个中断源，2个优先级。

STC12C5A60S2的中断系统有10个中断源，2个优先级或4个优先级。可实现二级或四级中断服务嵌套。(详见数据手册P57)

可以分为内部中断源和外部中断源

一次中断过程的完整步骤

- 1、中断请求：中断事件一旦发生，中断源就提交中断请求（将中断标志位置1），欲请求CPU暂时放下目前的工作转向为该中断作专项服务。
- 2、中断使能：虽然中断源提交了中断请求，但是，能否得到CPU的响应，还要取决于该中断请求能否通过若干关卡送达CPU（中断使能位等于1，关卡放行），这些关卡有以下两类：
 - 此中断源的中断允许位；
 - 全局中断允许位。
- 3、中断响应：如果一路放行，则CPU响应该中断请求，记录断点，跳转到中断服务程序。对于INT和TMR中断，中断响应时中断标志位会被硬件自动清零。
- 4、中断处理：对中断源进行有针对性的服务。
- 5、中断返回：返回到主程序断点处，继续执行主程序。

1、3、5由硬件自动完成。2、4是用户编程完成。

中断响应条件为1、2同时满足。



中断请求标记的置位和清除

外部中断

下降沿触发方式条件下，在产生中断请求时由硬件置位（置1）中断请求标记，当CPU响应中断时由硬件清除（清0）。

电平触发方式条件下，中断请求标记由外部中断源控制。具体是：当CPU检测到INT引脚上出现低电平时，中断标志IE由硬件置位，INT引脚上出现高电平时，中断标志IE由硬件清除。

定时器中断

计数溢出时由硬件置位中断请求标记，当CPU响应中断由硬件清除。

串口中断

当串行口接收完一帧数据后请求中断，由硬件置位中断请求标记RI，RI必须由软件清除。

当串行口发送完一帧数据后请求中断，由硬件置位中断请求标记TI，TI必须由软件清除。



每个中断源的3个位

优先级定义位：每个中断源可以选择中断优先权（Priority二选一或四选一）

中断使能位：每个中断源均可使能（Enable）或使不能(Disable)

中断请求标记位：每个中断源均有独立的中断请求标记（Flag）

中断请求标记的产生：中断事件发生时，由硬件自动产生

中断请求标记的清除：中断事件被响应时，由硬件自动清除



全局中断使能位

EA：全局中断使能位（总允许位）。

=0：全局中断禁止；=1：全局中断允许。



外部中断

中断源在CPU的外部，称为外部中断 (**external interrupt**)。

1T单片机的外部中断最多可以达到4个，2个和传统51单片机完全一致，为INT0、INT1。

另外2通道捕获/比较单元（PWM/PCA/CCP，第4、5管脚），也可用来再实现2个外部中断（支持上升沿/下降沿中断）。

本讲只讲和传统51单片机的2个外部中断，如果需要后2个外部中断，可参考STC12C5A60S2数据手册P156

[例程7-1]:主程序为1位数码管上进行0—F的自动计数。中断服务程序：按下SW0，发出“嘀”一声。

中断系统的组成：由硬件系统和软件系统两部分组成。先进行硬件系统分析。



每个中断源的3个位:

PXn: (**P**riority)外部中断优先级定义位。

- =0: 低优先级; =1: 高优先级;
- 也可不设置, 上电时, 默认=0: 低优先级。

EXn: 外部中断使能位。

- =0:使不能; =1: 使能;
- 上电时, 默认=0:使不能

IEn: 外部中断请求标记位。

- =0:无中断请求; =1: 有中断请求;
- 上电时, 默认=0:无中断请求。

与所有中断源相关的1个位:

EA (Enable All): 全局中断使能位。

- =0:使不能; =1: 使能;
- 上电时, 默认=0:使不能



INT触发方式控制位

*INT*中断多了1个需要设置的位:

ITn(INT Trigger)外部中断触发方式控制位。

- =0:电平触发方式; =1:下降沿触发方式
- 也可不设置, 上电时, 默认=0:电平触发方式。

弟子教你学51单片机技术

中断服务函数



定义中断函数的一般形式:

`void 函数名() interrupt n`

- (1) `interrupt`必须要加, 表示定义成中断服务函数。
- (2) 后面`n`是中断号, 中断号是编译器识别不同中断的唯一编号。

中断服务函数和普通函数的异同:

同: 函数的形式非常类似, 中断响应过程和普通函数调用过程也非常相似。

异:

- 中断服务函数不需要声明, 普通函数一般需要声明。
- 普通函数的执行是可预测的; 而中断服务函数的执行是不可预测的。
- 普通函数的跳转是软件(函数调用语句)完成的; 中断的跳转(中断响应)是由硬件完成的, 只要发生了中断事件, 并且中断被允许, 硬件会自动帮我们完成中断的跳转(中断响应)
- 普通函数: 通过函数名找到被调用函数; 中断服务函数: 通过中断号找到中断服务函数。所以我们可以知道, 中断服务函数中的函数名其实并没有什么作用。

使用中断函数时应遵循以下规则:

- (1) 中断函数不能进行参数转递;
- (2) 在任何情况下, 都不能直接调用中断函数;

从零到项目开发学51单片机与C语言技术视频教程



中断的优先级

如果多个中断源同时提出了中断请求，怎么办？

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。注意：这个查询是硬件自动帮我们完成的，程序员并不需要为此书写语句。



2.10.1 中断优先级

STC12C5201AD 系列 STC12C5A60S2 系列单片机 中断优先级及中断查询次序，与 8051 完全兼容

Interrupt Source 中断源	Vector Address 中断向量地址	Polling Sequence 中断查询次序	中断优先级设置 (IPH, IP)	优先级0 最低	优先级1	优先级2	优先级3 最高	Interrupt Request 中断请求标志位	Interrupt Enable Control Bit 中断允许控制位
/INT0	0003H	0(最优先)	PX0H, PX0	0, 0	0, 1	1, 0	1, 1	IE0	EX0 / EA
Timer 0	000BH	1	PT0H, PT0	0, 0	0, 1	1, 0	1, 1	TF0	ET0 / EA
/INT1	0013H	2	PX1H, PX1	0, 0	0, 1	1, 0	1, 1	IE1	EX1 / EA
Timer 1	001BH	3	PT1H, PT1	0, 0	0, 1	1, 0	1, 1	TF1	ET1 / EA
UART	0023H	4	PSH, PS	0, 0	0, 1	1, 0	1, 1	RI + TI	ES / EA
ADC	002BH	5	PADCH, PADC	0, 0	0, 1	1, 0	1, 1	ADC_FLAG	EADC / EA
LVD	0033H	6	FLVDH, PLVD	0, 0	0, 1	1, 0	1, 1	LVDF	ELVD / EA
PCA	003BH	7	PPCAH, PPCA	0, 0	0, 1	1, 0	1, 1	CF + CCF0 + CCF1	(ECF+ECCF0+ECCF1) / EA
以上是12C5202AD/12C5202PWM/12C5202系列和12C5A60S2/AD/PWM系列共有的中断									
UART2	0043	8	PS2H, PS2	0, 0	0, 1	1, 0	1, 1	S2TI + S2RI	ES2 / EA
UART2是12C5A60S2系列独有的第二个串口中断(12C5A60AD系列/12C5A60PWM系列/12C5202AD系列没有UART2)									
SPI	004B	9	PSPIH, PSPI	0, 0	0, 1	1, 0	1, 1	SPIF	ESPI / EA
SPI是12C5A60S2/12C5A60AD/12C5A60PWM系列系列系列独有的SPI中断									



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会如何处理？

如果新的中断优先级与当前正在处理的中断是同级的，则不予响应，待当前中断服务程序执行完毕后，再响应。

如果新的中断优先级比当前正在处理的中断高的，则会发生中断嵌套。

结论：优先级是程序员指定的。在多中断源程序的编写中，程序员必须认真考虑优先级问题，否则中断系统会运行不正常，甚至导致危险的发生。

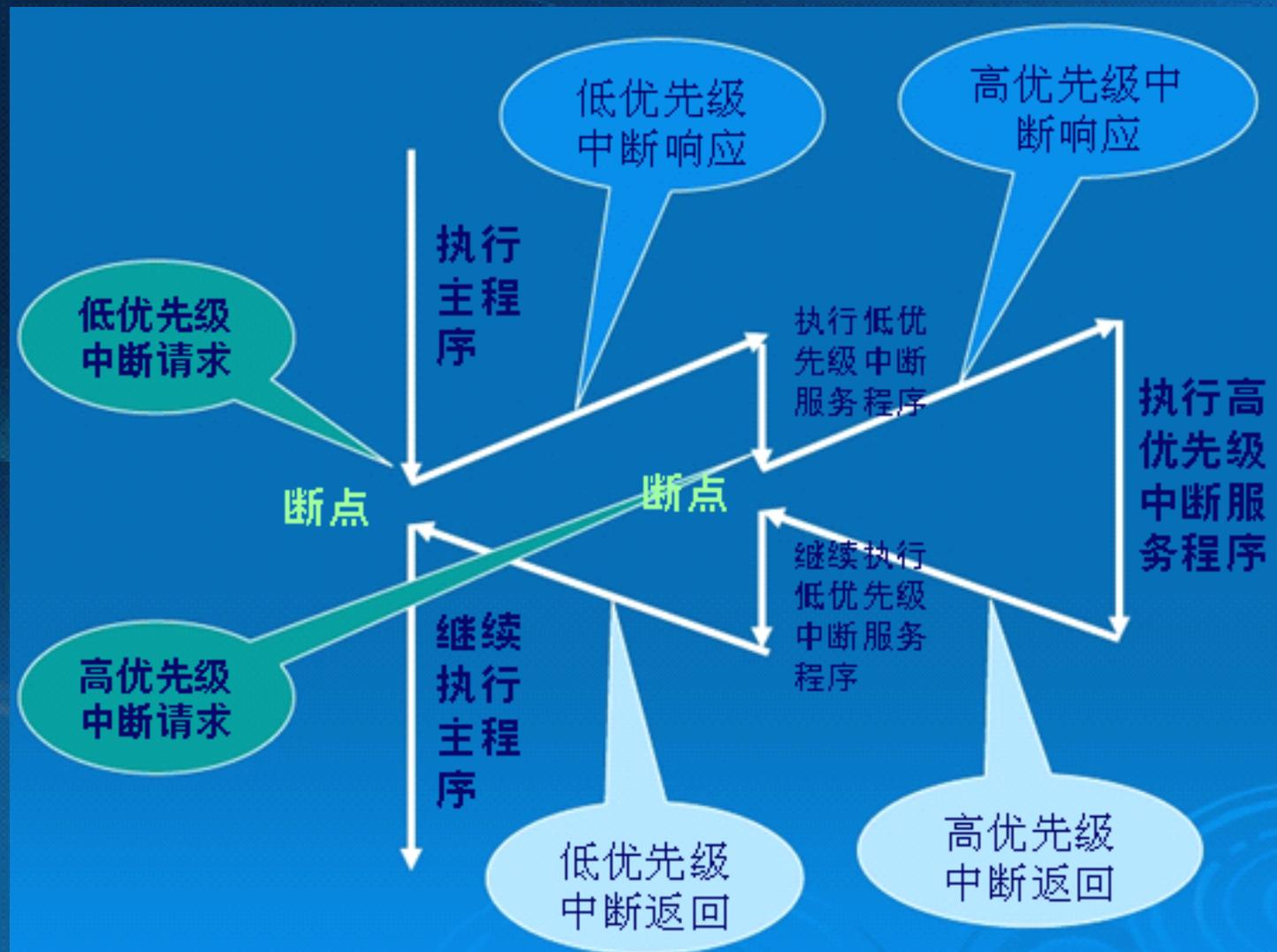


简单中断程序

```
• #include<reg52.h> //包含头文件，一般情况不需要改动，头文件包含特殊功能寄存器的定义
• /*-----
•                               主程序
• -----*/
• main()
• {
•     P1=0x55;      //P1口初始值
•     EA=1;        //全局中断开
•     EX0=1;       //外部中断0开
•     IT0=0;       //电平触发
•     while(1)
•     {
•         //在此添加其他程序
•     }
• }
• /*-----
•                               外部中断程序
• -----*/
• void ISR_Key(void) interrupt 0 using 1
• {
•     P1=~P1;      //进入中断程序执行程序，
•     //此时可以通过 EA=0指令暂时关掉中断
• }
```



中断的嵌套





中断优先级的三条原则：

CPU同时接收到几个中断时，首先响应优先级别最高的中断请求。

正在进行的中断过程不能被新的同级或低优先级的中断请求所中断。

正在进行的低优先级中断服务，能被高优先级中断请求所中断。



课后作业

- 1、让蜂鸣器发出**500HZ**声音，要求利用定时器**0**实现。
- 2、输出周期为**1s**的方波，让点阵管上某发光点以**1HZ**闪烁，要求利用定时器**0**实现。
- 3、主程序为**1**位数码管上进行**0—9**的自动计数。按下**SW0**，停止计数，再按下**SW0**恢复计数。要求用定时器**0**和外部中断编程实现。