
项目八 外部中断控制

陈超然

汕头职业技术学院

目录

1. 认识中断系统和外部中断

- 什么是中断
- 中断源
- 什么是外部中断

2. 外部中断的控制

- 认识RAM
- 控制外部中断的管理员
- 设置外部中断的具体步骤

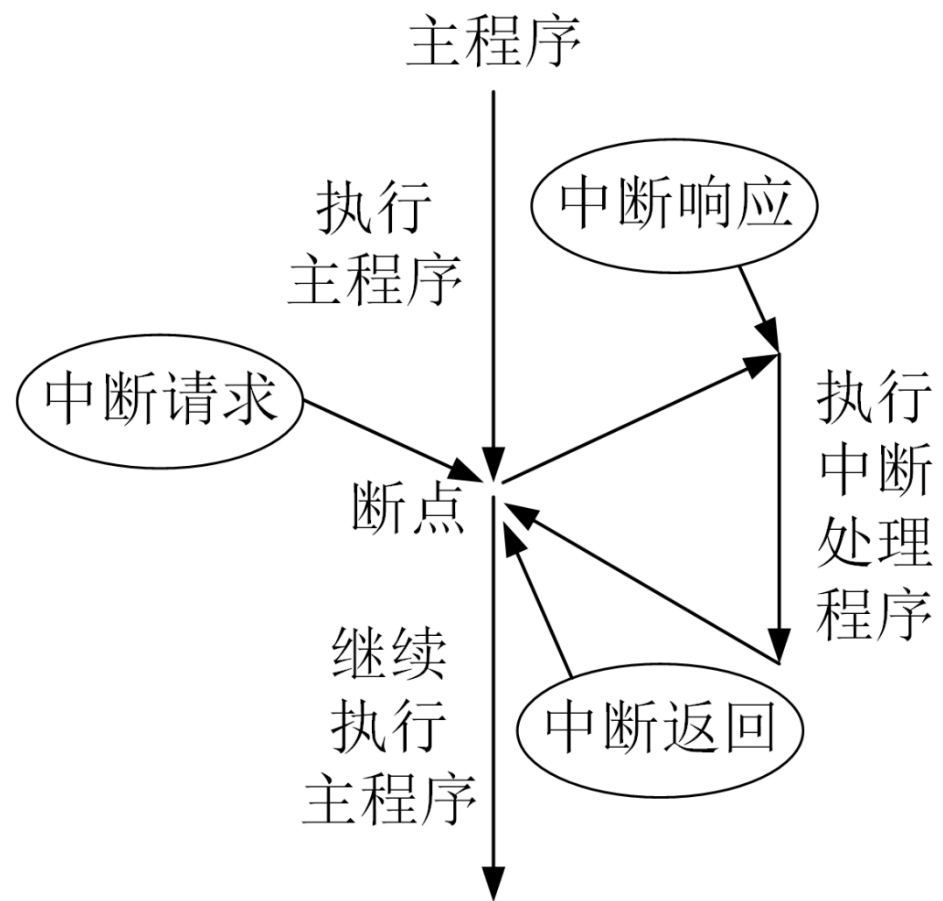
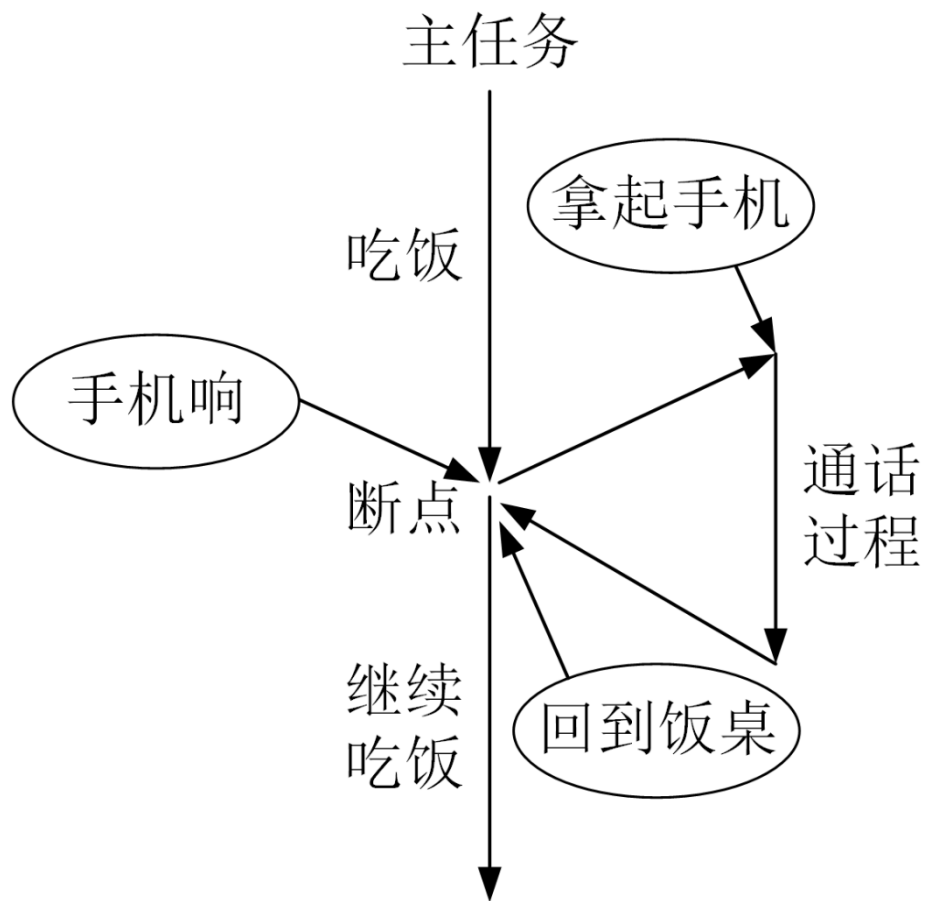
3. 实战任务

- 任务要求和分析
- 仿真软件练习
- 开发板练习

4. 实战作业

8.1 认识中断系统和外部中断

1、什么是中断

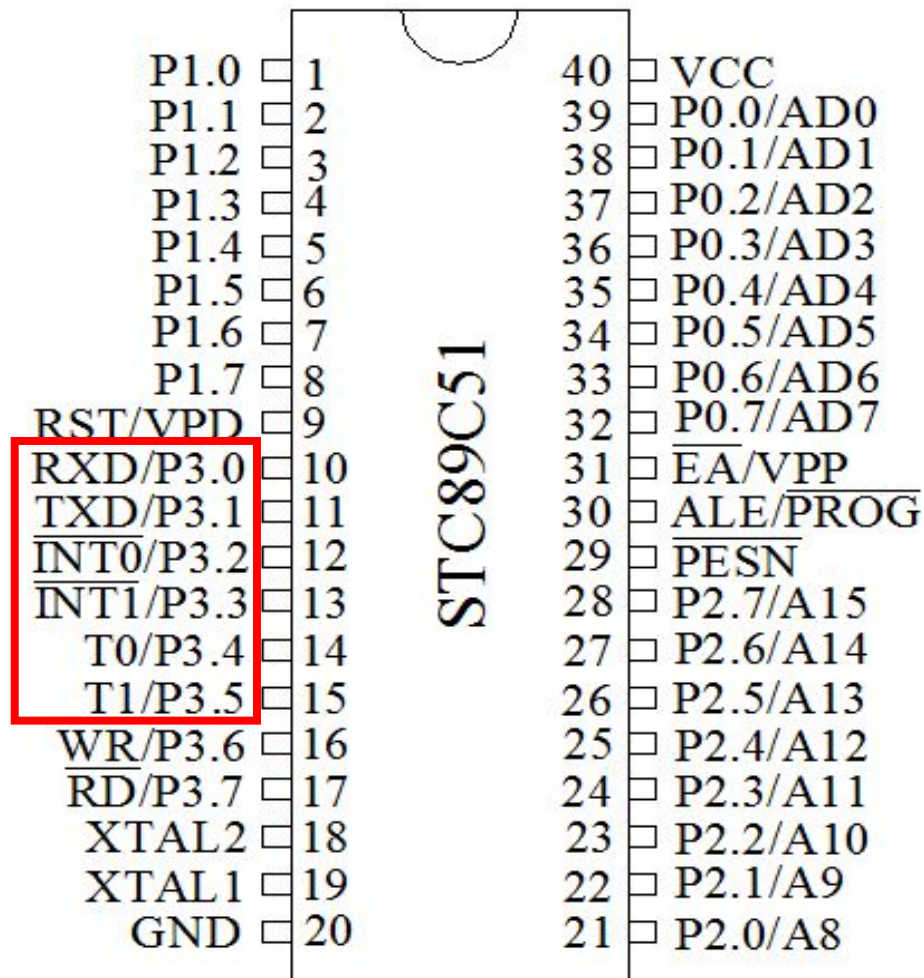


8.1 认识中断系统和外部中断

2、中断源

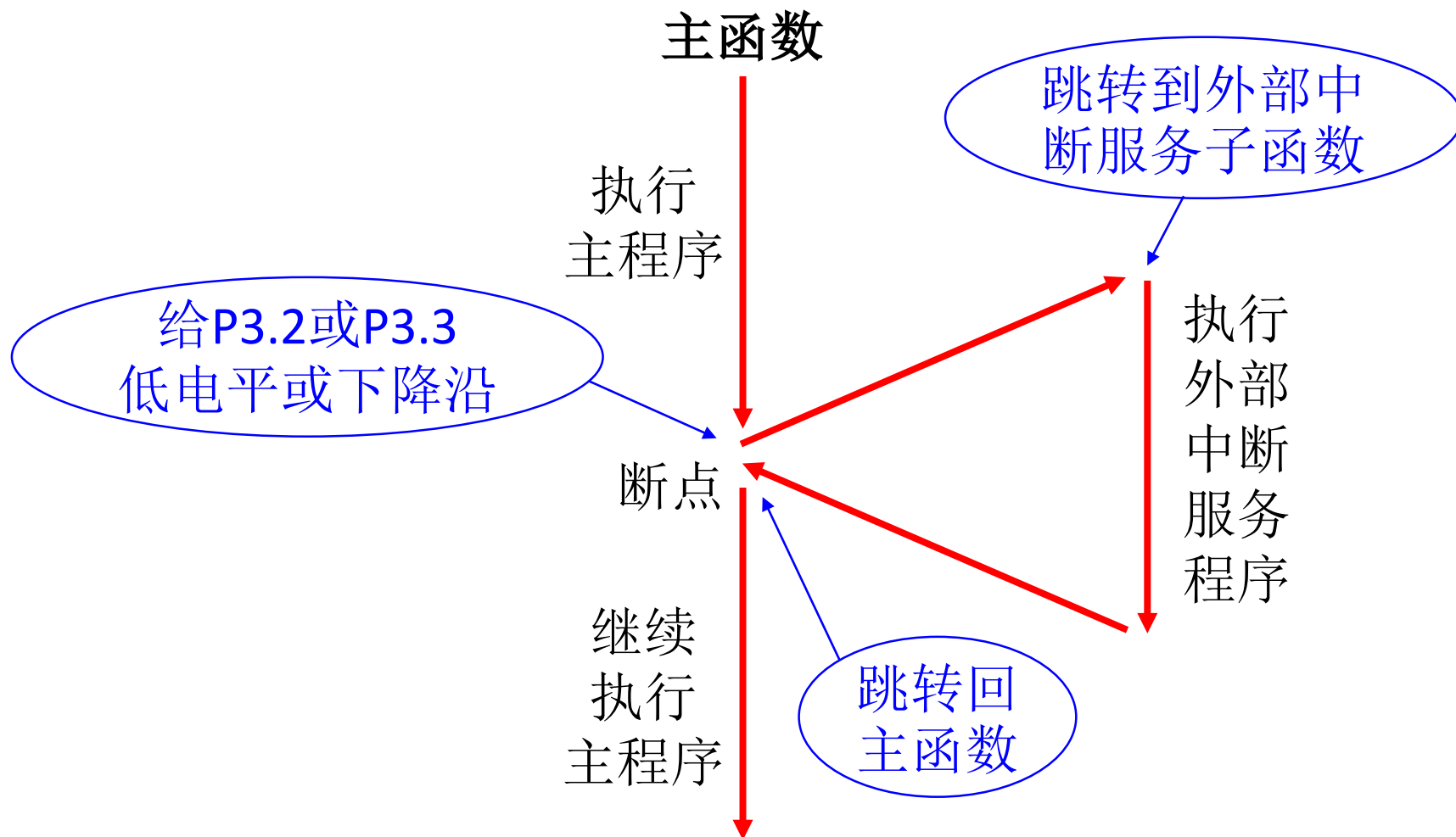
• 51单片机有5个中断源：

1. 外部中断0 ($\overline{\text{INT0}}$ / P3.2)
2. 定时/计数器0 (T0 / P3.4)
3. 外部中断1 ($\overline{\text{INT1}}$ / P3.3)
4. 定时/计数器1 (T1 / P3.5)
5. 串行口中断 (RXD/P3.0、TXD/P3.1)



8.1 认识中断系统和外部中断

3、什么是外部中断



单片机芯片主要包括以下哪些？

- ☒ A 中央处理器
- ☒ B 存储器
- ☒ C 输入/输出接口电路
- ☐ D 显卡和声卡

提交

8.2 外部中断的控制

复习回顾（过渡）

【填空】单片机芯片内部的存储器可分为____和____？

【答案】程序存储器ROM 和 数据存储器RAM

- 提问：51单片机的ROM和RAM分别多大呢？
- ROM为4KB
RAM为256B

小米10

科技突破极限

骁龙865 | 新一代LPDDR5内存 | 1亿像素8K电影相机 | 对称立体声



内存与容量

12GB + 256GB

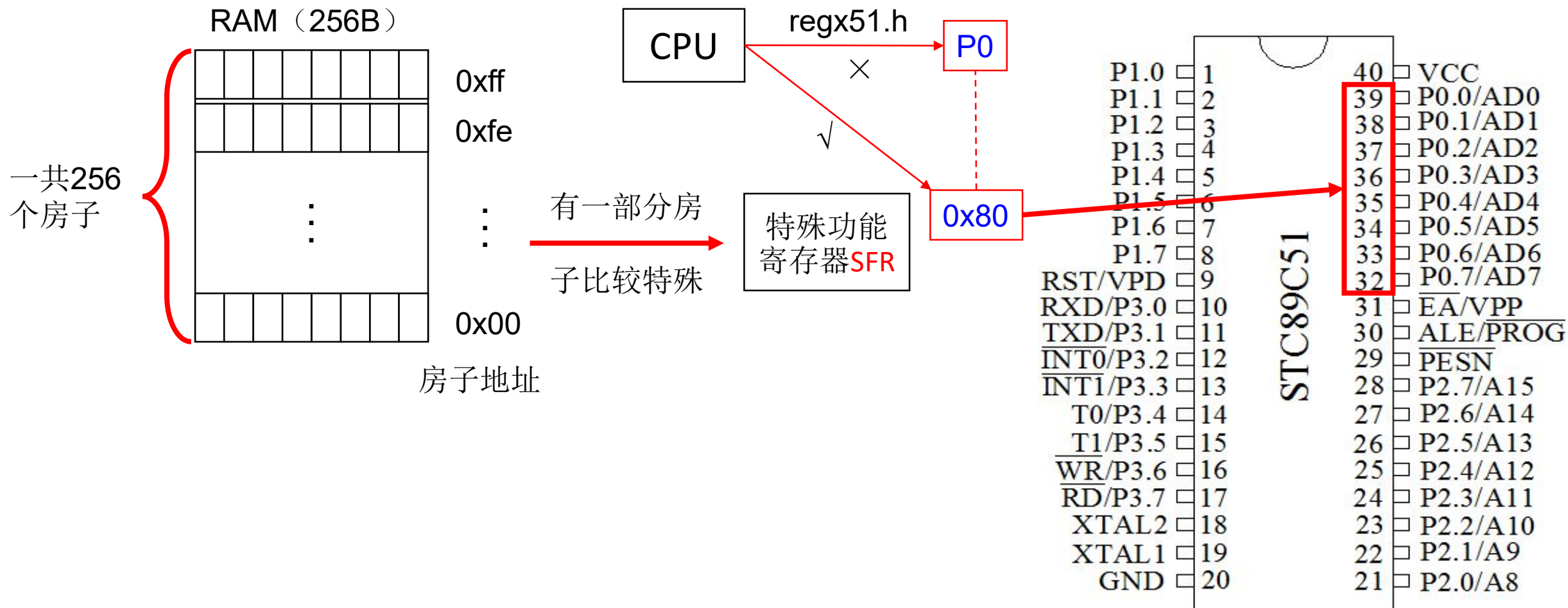
最高可选

运行内存：8GB / 12GB	LPDDR5 高速内存
机身存储：128GB / 256GB	UFS 3.0 高速存储

*实际可用容量会由于诸多因素而减少并有所差异：由于操作系统运行占据了部分内存（RAM），实际可用空间小于标识内存容量；
由于安装操作系统和预装的程序占据了部分闪存（ROM），实际可用存储空间小于标识闪存容量。

8.2 外部中断的控制

1、认识RAM



8.2 外部中断的控制

2、控制外部中断的管理员

IE:
(0xa8)

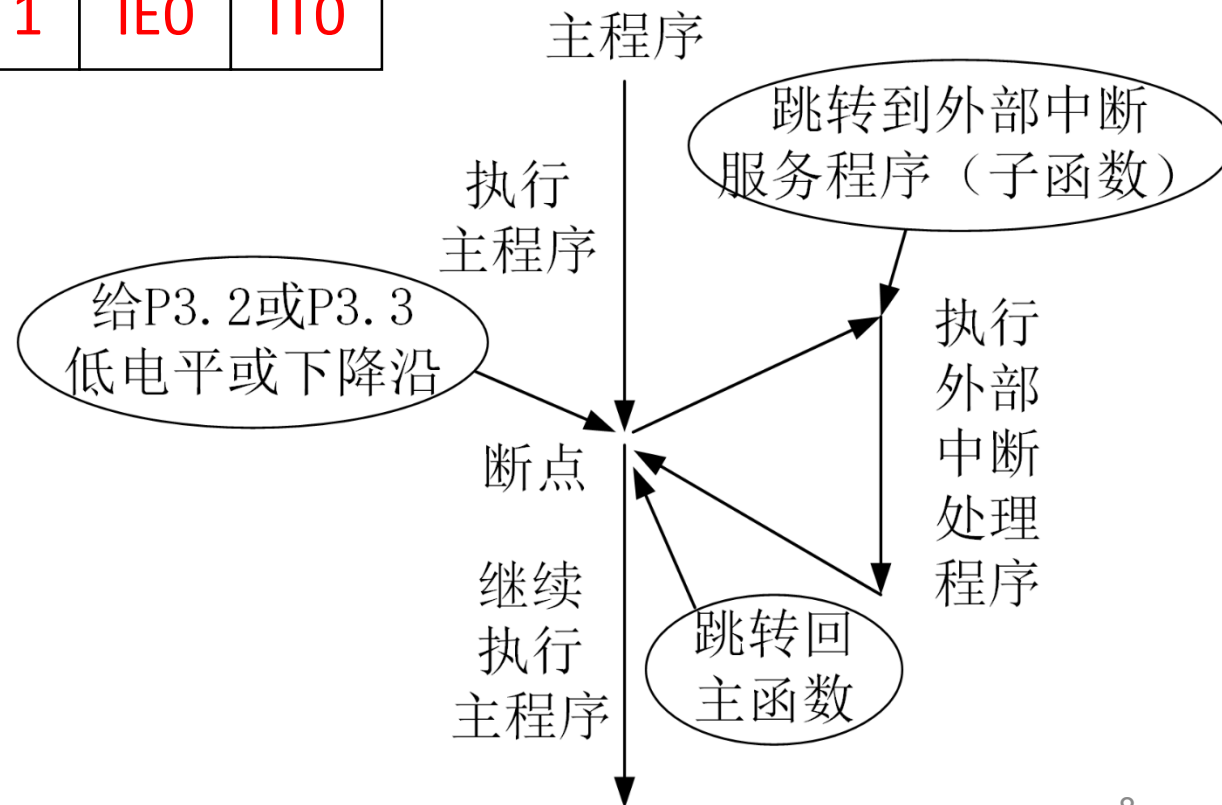
EA	-	-	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

TCON:
(0x88)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

EA=1;
EX0=1;
IT0=1;

- EA: 总中断 允许位
- EX0: 外部中断0 允许位
- EX1: 外部中断1 允许位
- IT0: 外部中断0触发方式 选择位
- IT1: 外部中断1触发方式 选择位
- IE0: 外部中断0请求 标志位
- IE1: 外部中断1请求 标志位



8.2 外部中断的控制

3、设置外部中断的具体步骤

1. 硬件(接线):

外部中断0: P3.2、外部中断1: P3.3

2. 软件(编程):

① 外部中断初始化:

IE { a) 打开总中断: EA=1;

b) 打开外部中断0/1: EX0/EX1=1;

TCON { c) 选择外部中断0/1的触发方式: IT0/IT1=0; (低电平触发)
IT0/IT1=1; (下降沿触发)

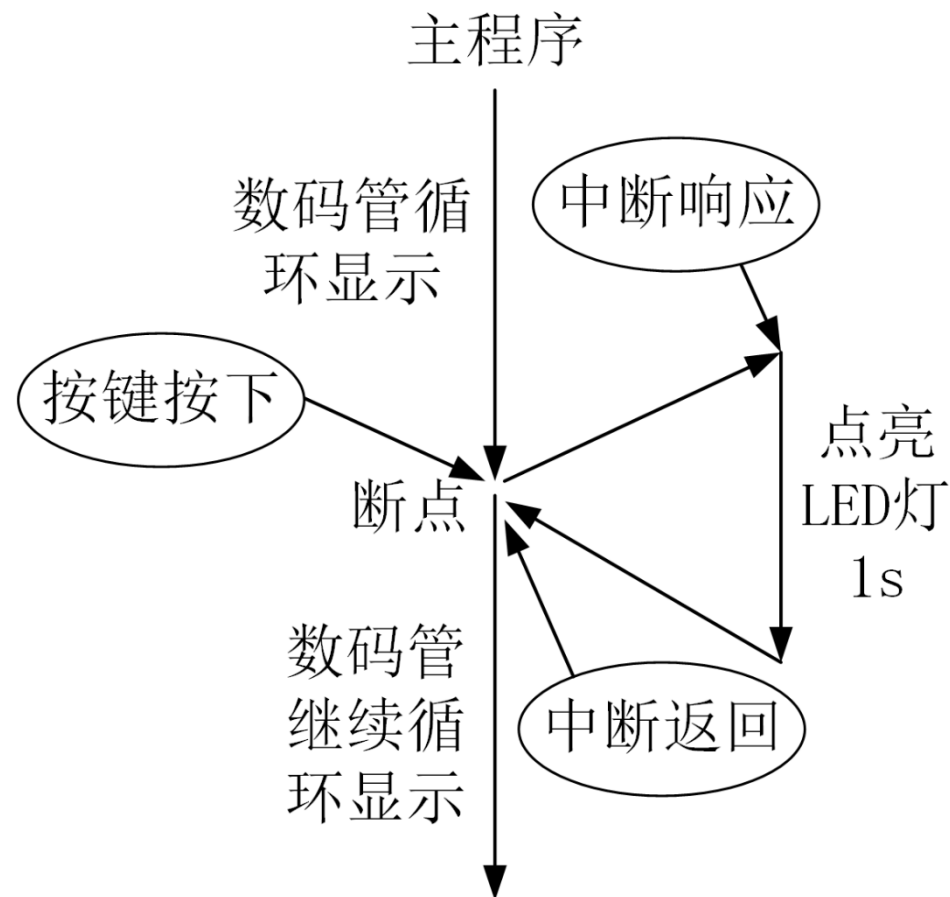
② 外部中断0/1中断服务程序 (子函数) :

```
void xxx() interrupt 0/2  
{.....}
```

8.3 实战任务

按键外部中断数码管循环显示

- 任务要求:
- 单片机一通电，数码管开始间隔0.5秒循环加一显示0~9
- 当按下按键，则触发外部中断，数码管停止加一显示，同时点亮LED灯1秒（此时数码管保持显示停止时的数字），之后自动熄灭LED并继续循环加一显示0~9
- 任务分析:
- 用外部中断0来做，则按键接P3.2口。为避免多次触发外部中断，采用下降沿触发的方式。即无论按键按多久，都只进入触发一次中断



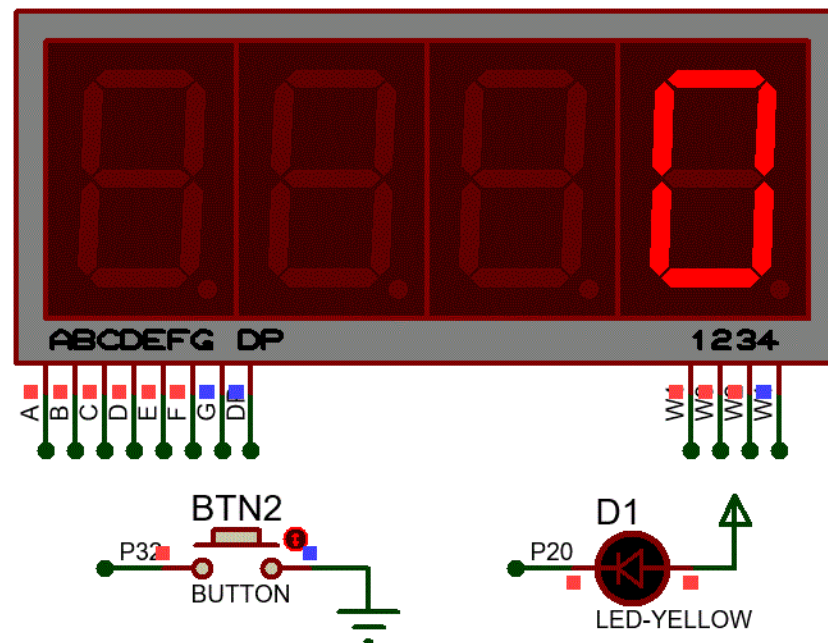
仿真步骤1：画电路图（与开发板相同）



8.3 实战任务

仿真步骤2：编程 & 仿真

```
#include <regx51.h>
#define uint unsigned int
#define uchar unsigned char
//共阴极数码管
uchar code table[]={0x3f,0x06,0x5b,
0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,
0x77,0x7c,0x39,0x5e,0x79,0x71};
void delay(uint i);
void main(void)
{
    uchar i;
    //外部中断0初始化
    EA=1;          //开总中断
    EX0=1;         //开外部中断0
    IT0=1;         //外部中断0的触发方式为下降沿触发
    //位选：选通最右边数码管
    P2_2=0; P2_3=0; P2_4=0;
    while(1)
    {
        for(i=0;i<10;i++)
        {
            P0=table[i];
            delay(55500);
        }
    }
}
```

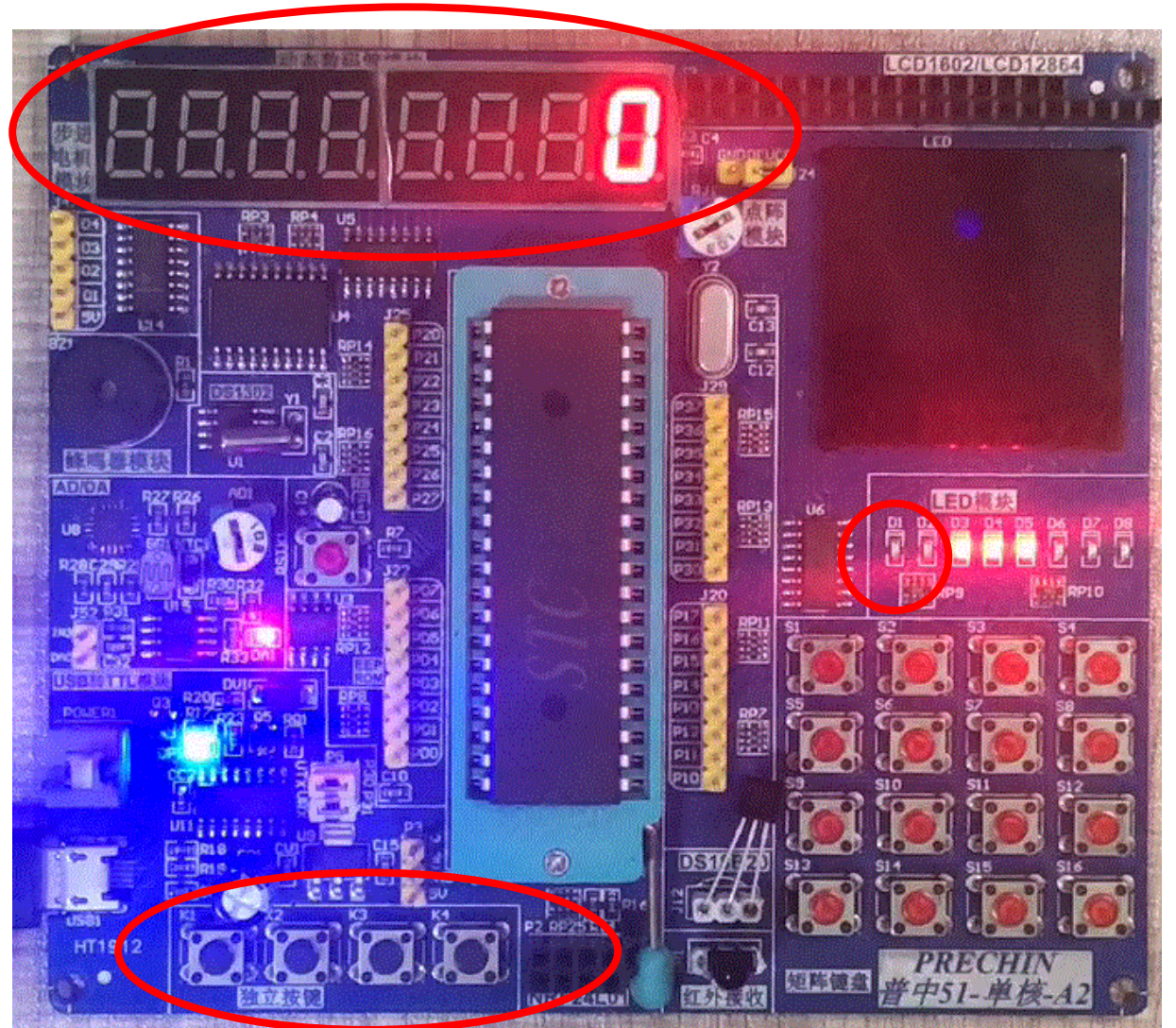


```
//延时子函数
void delay(uint i)
{
    while(i--);
}
//外部中断0子函数：点亮led灯1秒
void exter0() interrupt 0
{
    P2_0=0;
    delay(55500); //延时0.5秒
    delay(55500); //延时0.5秒
    P2_0=1;
}
```

8.3 实战任务

开发板练习

- 由于仿真练习的电路图设计与实际开发板原理图相同，故直接将用于仿真的程序直接下载到实际开发板，观察实际效果如右图。



8.4 实战作业

1. 用外部中断0做计时器：

- 数码管的显示范围为0~F，且初始显示为0；
- 当按一次按键，数码管开始自动每秒加1，并循环显示；
- 再按一次按键，数码管停止自动加1，并保持显示原数；

2. 在第1题基础上，用外部中断1继续完善计时器：

- 按下第二个按键，计时器清零从头开（如果数码管清零前正自动每秒加1，则清零后依旧自动每秒加1）