单片机原理及应用

项目六：手动计数器设计

1. 理解中断的基本概念；
2. 熟悉单片机的中断寄存器；
3. 掌握单片机中断系统的结构与原理；
4. 掌握中断的使用方法；
5. 掌握单片机 C 语言中断控制设计方法；
6. 掌握单片机外部中断脉冲触发方式使用。

1. 会阐述中断控制电路的工作原理及控制方法单片机的定时 / 计数器的结构、存储器；
2. 会初始化外部中断的寄存器；
3. 会使用单片机控制外部寄存器的编程方法；

1. 能分析设计任务，正确使用外部中断；
2. 能熟练编写单片机控制外部中断；
3. 能使用 PROTUES 软件绘制电路原理图；
4. 能使用 Keil 软件编译程序对外部中断的控制，并与 PROTUES 软件联调，实现控制电路仿真；
5. 能够完成手动计数器设计与仿真；
6. 能够使用外部中断去解决实际问题。

4 课时

1.1 任务提出

设计一个手动 0 ～ 99 计数器，记录按键的次数。系统上电时，数码管显示 00 ，利用单片机引脚外接一个按键，作为计数器，计数从 0 开始，到 99 后变为 0 ，如此循环，利用数码管显示计数结果。

1.2 方案设计

1. 单片机选择。采用 ATMEL 公司的 AT89C51 作为系统控制器的 CPU 方案。单片机算术运算功能强，软件编程灵活、自由度大，可以用软件编程实现各种算法和逻辑控制，并且由于其功耗低、体积小、技术成熟和成本低等优点，使其在各个领域应用广泛。

2. 显示电路的选择。根据题目要求，能够直观显示倒计时时间。采用 LED 数码管显示， LED 数码管价格适中，显示控制也方便，显示的方式采用动态显示方法。由于显示的范围是 0 到 99 ，因此采用一个两位一体的共阳极数码管。

3. 计数输入电路的选择。外接一个按键，获取按键按下的次数。通常采用查询的方法，当检测到按键按下，就计 1 （加 1 ），通过单片机处理后送显示电路显示。也可以采用外部中断的方式，当按键按下，就产生一次中断，计数值 1 （加 1 ），通过单片机处理后送显示电路显示。本系统采用的是中断方式。

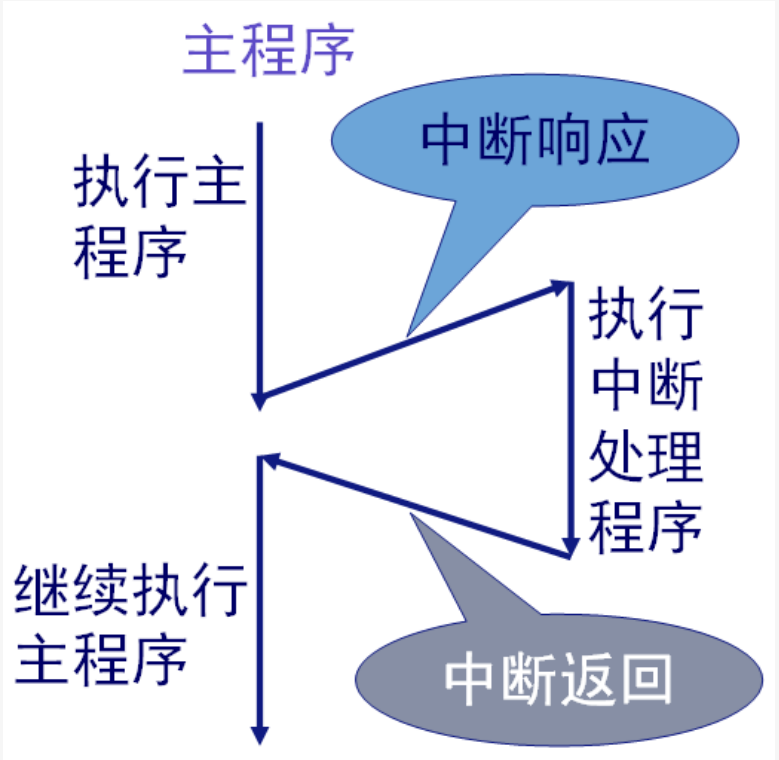
1.3 硬件设计

1. 中断

在单片机中，当 CPU 在执行程序时，由单片机内部或外部的原因引起的随机事件要求 CPU 暂时停止正在执行的程序，而转向执行一个用于处理该随机事件的程序，处理完后又返回被中止的程序断点处继续执行，这一过程就称为中断。

51 系列单片机共有 5 个中断源。它们分别是： 3 个内部中断，即定时器 T0 的溢出中断、定时器 T1 的溢出中断和串行口中断； 2 个外部中断即 $\overline{INT0}$(P3.2) 和 $\overline{INT1}$(P3.3)。可以根据需要随时向 CPU 发出中断申请。当外部中断源超过两个时，还可以通过一定的方法扩充。

引入中断可以提高 CPU 工作效率， CPU 不必花费大量时间等待和查询外设工作；能够实现实时处理功能，中断源根据外界信息变化可以随时向 CPU 发出中断请求，若条件满足， CPU 会马上响应，对中断要求及时处理；能够实现分时操作。



1.3 硬件设计

1.1 中断步骤

1 ）中断请求。向 CPU 发出中断请求的来源，或引起中断的原因称为中断源，可分为软件中断和硬件中断，也分为内部中断源和外部中断源。中断源要求服务的请求称为中断请求。当中断源请求 CPU 中断时，会向 CPU 提出中断请求，中断请求信号产生一次， CPU 只响应一次， CPU 不可能响应多次的现象。

2 ）中断响应。 CPU 对系统内部中断源提出的中断请求必须响应，而且自动取得中断服务子程序的入口地址，执行中断服务子程序。对于外部中断， CPU 在执行当前指令的最后一个时钟周期去查询外部中断引脚，若查询到中断请求信号有效，同时在系统开中断的情况下， CPU 向发出中断请求的外设回送一个低电平有效的中断应答信号，作为对中断请求的应答，系统自动进入中断响应周期。单片机中断响应过程是：

a. 对相应的优先级状态触发器由硬件置 1 。

b. 保护断点。

c. 清除中断请求标志，例如 IE0 ， IE1 ， TF0 ， TF1 。

d. 关闭同级中断。在一种中断响应后，同一优先级的中断被暂时屏蔽，待中断返回时再重新打开。

e. 将相应中断的入口地址送入 PC 。

1.3 硬件设计

1.1 中断步骤

3）中断处理。中断处理是执行中断的主体部分，不同的中断请求，有各自不同的中断服务内容，需要根据中断源所要完成的功能，事先编写相应的中断服务子程序存入内存，等待中断请求响应后调用执行。中断服务一般包含以下几个部分：

① 保护现场。所谓保护现场，就是指在中断响应时，将断点处的有关寄存器的内容（如特殊功能寄存器 ACC 、 PSW 、 DPTR 等）压入堆栈中保护起来，以便中断返回时恢复。

② 执行中断服务程序，完成相应操作。中断服务程序中的操作和功能是中断源请求中断的目的，是 CPU 完成中断处理操作的核心和主体

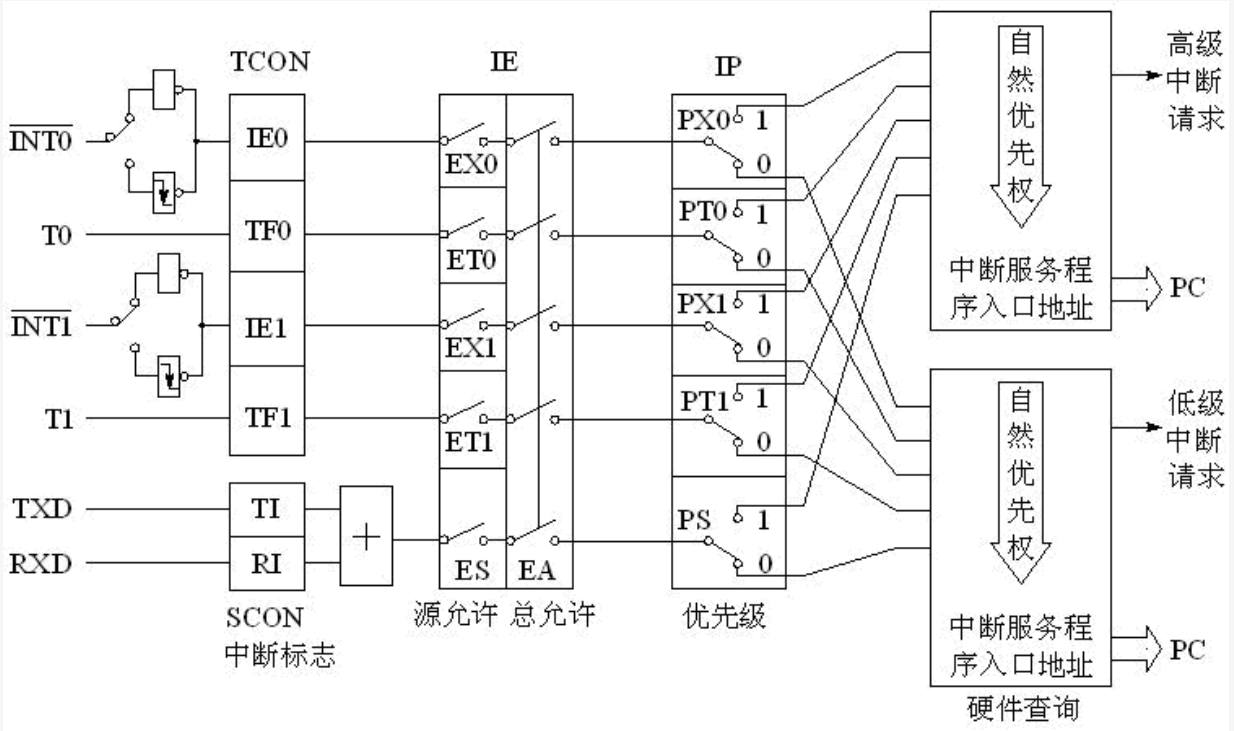
③ 恢复现场。与保护现场相对应，中断返回前，应将保护现场时压入堆栈中的相关寄存器中的内容从堆栈中取出，返回原有的寄存器，以便中断返回时继续执行原来的程序不出差错。

4）中断返回。中断返回是指中断服务完后，计算机返回原来断开的位置 (即断点) ，继续执行原来的程序。

1.3 硬件设计

1.2 中断结构

单片机的中断系统的内部结构图如图 6-2 所示，中断系统有 5 个中断请求源和 4 个用于中断控制的寄存器。定时控制寄存器（ TCON ）、串行控制寄存器（ SCON ）、中断控制寄存器（ IE ）和中断优先级控制寄存器（ IP ）来控制中断的类型、中断的开关和各种中断源的优先级。

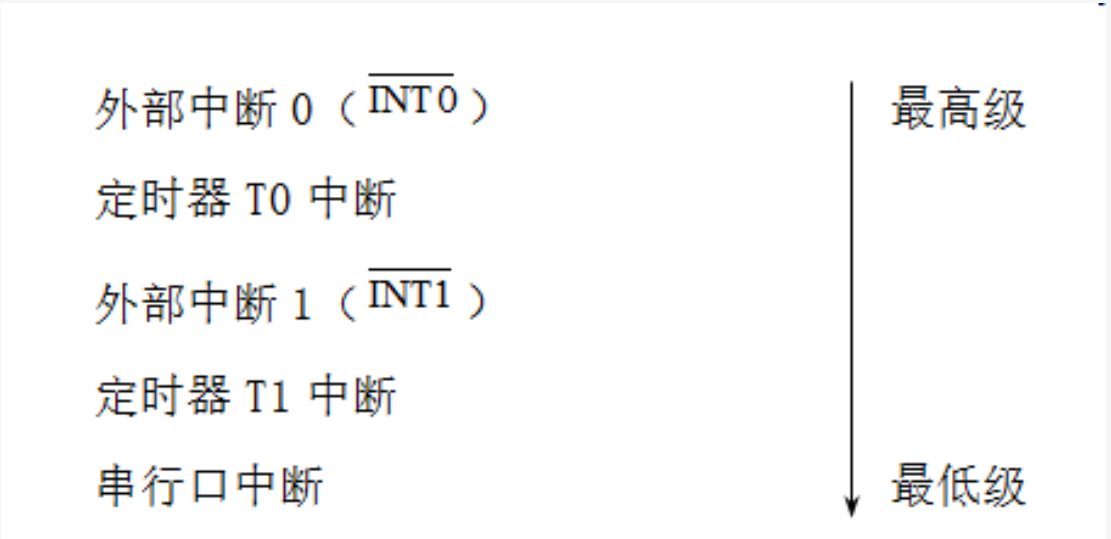


1.3 硬件设计

1.3 中断优先级

当几个中断源同时请求中断，或者当某一个中断正在响应中 (即正在执行该中断源的中断服务程序), 又有其它的中断源请求中断 , 这时中断系统应如何处理呢 ? 这就涉及到中断优先级的问题。单片机的中断系统 , 只规定了两个中断优先级：高优先级中断或低优先级中断，需要通过指令预先设置完成。

若同一级有几个中断源申请，则按自然优先级顺序响应。



1.3 硬件设计

1.3 中断优先级

对于中断优先级和中断嵌套， 51 单片机有以下三条规定。

① 正在进行中的中断过程不能被新的低优先级的中断请求所中断，直到该中断服务程序结束，返回主程序且执行了主程序中的一条指令后， CPU 才响应新的中断请求。

② 正在进行的低优先级中断服务程序能被高优先级中断请求所中断，实现两级中断嵌套。

③ CPU 同时接收到几个中断请求时，首先响应优先级最高的中断请求。

实际上， 51 单片机对于二级中断嵌套的处理是通过中断系统中的两个用户不可寻址的优先级状态触发器来实现的。这两个优先级状态触发器是用来记录本级中断源是否正在中断。如果正在中断，则硬件自动将其优先级状态触发器置“ 1”。若高优先级状态触发器置“ 1”，则屏蔽所有后来的中断请求。若低优先级状态触发器置“ 1”，则屏蔽所有后来的低优先级中断，允许高优先级中断形成二级嵌套。当中断响应结束时，对相应的优先级状态触发器由硬件自动清零。

1.3 硬件设计

1.4 中断寄存器

1 ）中断标志寄存器 TCON

TCON （定时器控制寄存器）的作用是控制定时器的启动与停止，并保存 T0 、 T1 的溢出中断标志和外部中断、的中断标志。

TCON	D7	D6	D5	D4	D3	D2	D1	D0
位名称	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

① TF1 为定时器 T1 溢出中断请求标志位。当定时器 / 计数器 T1 计数溢出后，由 CPU 内硬件自动置“ 1”，表示向 CPU 请求中断。 CPU 响应该中断后，内部硬件自动对其清 0 。 TF1 也可由软件程序查询其状态或软件置位清 0 。
② TF0 为定时器 T0 溢出中断请求标志位。其意义和功能与 TF1 相似。
③ IE1 为外部中断中断请求标志位。当 P3.3 引脚信号有效时，触发 IE1 置“ 1”， 当 CPU 响应该中断后。由内部硬件自动清 0 （自动清 0 只适用于边沿触发方式）。
④ IE0 为外部中断中断请求标志位。其意义和功能与 IE1 相似。
⑤ IT1 为外部中断触发方式控制位。 IT1=1 ，边沿触发方式，当 P3.3 引脚出现下跳变脉冲信号时有效 ； IT1=0 ，电平触发方式，当 P3.3 引脚为低电平时有效。 IT1 由软件置位或复位。
⑥ IT0 为外部中断触发方式控制位。其意义和功能与 IT1 相似。
⑦ TR1 ： 定时器 1 启停控制位； TR0 ： 定时器 0 启、停控制位。其功能同 TR1 。

1.4 中断寄存器

2 ）中断标志寄存器 SCON

SCON （串行口控制寄存器）。 51 的串行口中断源对应两个中断标志位：串行口发送中断标志位 TI 和串行口接收中断标志位 RI 。无论哪个位置“ 1”。都请求串行口中断。到底是发送中断 TI 还是接受中断 RI ，只有在中断服务程序中通过指令查询来判断。串行口中断响应后，不能由硬件自动清零，必须由软件对 TI 或 RI 清零。串行收发结束的中断标志位被锁存在串行控制寄存器 SCON 中，其内部结构形式如表 6-2 所示。

SCON	D7	D6	D5	D4	D3	D2	D1	D0
位名称	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

① TI ：串行发送中断请求标志。 CPU 将一个字节数据写入发送缓冲器 SBUF 后启动发送，每发送完一帧数据，硬件自动使 TI 置 1 。但 CPU 响应中断后，硬件并不能自动使 TI 清 0 ，必须由软件使 TI 清 0 。

② RI ：串行接收中断请求标志。在串行口允许接收时，每接收完一帧数据，硬件自动使 RI 置 1 。但 CPU 响应中断后，硬件并不能自动使 RI 清 0 ，必须由软件使 RI 清 0 。

③ SM0 、 SM1 、 SM2 、 REN 、 TB8 和 RB8 等以后续串行口通信中详解。

1.3 硬件设计

1.4 中断寄存器

3）中断允许控制寄存器 IE

中断允许控制寄存器 IE 主要用于控制中断的开放和关闭。
单片机对各个中断源的中断允许和屏蔽是由内部的中断允许寄存器 IE 的各位来控制的。中断允许寄存器可以进行位寻址，各位的定义如表所示。

IE	D7	D6	D5	D4	D3	D2	D1	D0
位名称	EA	—	—	ES	ET1	EX1	ET0	EX0

① EA：中断允许总控制位。EA=0，屏蔽所有的中断请求；EA=1，开放中断。EA 的作用是使中断允许形成两级控制。即各中断源首先受 EA 位的控制；其次还要受各子中断源的中断控制。
② ES：串行口中断允许位。ES=0，禁止串行口中断；ES=1 允许串行口中断。
③ ET1：定时器 / 计数器的 T1 的中断允许位。ET1=0，禁止 T1 中断：ET1=1，允许 T1 中断。
④ EX1：外部中断 INT1 的中断允许位。EX1=0，禁止外部中断 INT1 中断；EX1=1，允许外部中断 INT1 中断。
⑤ ET0：计时器 / 计数器 T0 的中断允许位。ET0=0，禁止 T0 中断；ET0=1，允许 T0 中断。
⑥ EX0：外部中断 INT0 的中断允许位。EX0=0，禁止外部中断 INT0 中断；EX0=1 允许外部中断 INT0 中断。

1.3 硬件设计

1.4 中断寄存器

4 ）中断优先级控制寄存器 IP

中断优先级控制寄存器 IP 作用是设定各中断源的优先级别。
51 单片机有 5 个中断源，为了处理方便，每个中断源有两级控制：高优先级和低优先级。通过由内部的中断优先级寄存器 IP 来设置，中断优先级寄存器 IP 可以进行位寻址，各位定义见表。

IP	D7	D6	D5	D4	D3	D2	D1	D0
位名称	—	—	—	PS	PT1	PX1	PT0	PX0

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

② PT1：定时器 1 中断优先级控制位。 PT1=1 ，定时器 1 为高优先级中断； PTl=0 ，定时器 1 为低优先级中断。

③ PX1：外部中断 1 中断优先级控制位。 PX1=1 ，外部中断 1 为高优先级中断； PXl=0 ，外部中断 1 为低优先级中断。

④ PT0：定时器 0 中断优先级控制位。 PT0=1 ，定时器 T0 为高优先级中断 PT0=0 ，定时器 0 为低优先级中断。

⑤ PX0：外部中断 0 中断优先级控制位。 PX0=1 ，外部中断 0 为高优先级中断； PX0=0 ，外部中断 0 为低优先级中断。

1.3 硬件设计

1.5 中断服务程序

51 单片机中断是两级控制，在主程序中，要总中断允许，既令 EA=1；然后还要相应的子中断允许。在中断服务程序部分，要正确书写关键字 interrupt 和中断代码（见表 6-5 ）。中断服务程序的名字可任意，只要符合 C51 语法即可。

中断源名称		对应引脚	中断源服务程序入口
外部中断 0		INT0(P3.2)	0
定时器 / 计数器 0		T0(P3.4)	1
外部中断 1		INT1(P3.3)	2
定时器 / 计数器 1		T1(P3.5)	3
串行口中断	串行接收	RXD(P3.0)	4
	串行发送	TXD(P3.1)	

1.3 硬件设计

1.5 中断服务程序

1）中断函数的定义

在 C 语言中，中断函数使用关键词 interrupt 与中断号来定义中断函数，其一般形式如下：

```
Void 中断函数名（） interrupt 中断号 [using n]
{
        声明部分;
        执行部分;
}
```

格式说明：

中断函数无返回值，数据类型以 void 表示，也可省略。

中断函数名为标识符，一般以中断名称表示，力求简明易懂，如 T0_int( ) 。

中断号为该中断在 IE 寄存器的使能位置，如外部中断 0 的中断号为 0 。

选项 [using n] ，指中断函数使用的工作寄存器组号， n=0 ～ 3 。如果使用 [using n] 选项，编译器不产生保护和恢复 R0 ～ R7 ，执行会快一些，这时中断函数及其调用的函数必须使用不同的工作寄存器，否则会破坏主程序现场。而如果不使用 [using n] 选项，中断函数和主程序使用同一种寄存器，在中断函数中，编译器自动产生保护和恢复 R0 ～ R7 现场，执行速度会慢一些。一般情况下，主程序和低优先级中断函数使用同一组寄存器，而高优先级中断可用选项 [using n] 指定工作寄存器。

1.3 硬件设计

1.5 中断服务程序

2）中断函数的编写规则

不能进行参数传递。如果中断过程包括任何参数声明，编译器将产生一个错误信息。
无返回值。如果定义一个返回值将产生编译错误，但如果返回值的类型是默认的整型值，编译器将不能识别出该错误。

在任何情况下不能直接调用中断函数，否则编译器会产生错误。这是因为直接调用中断函数时，硬件寄存器上标志位没有中断请求存在，所以直接调用是不正确的。这就是中断函数和其他子函数的区别。

1.3 硬件设计

1.5 中断服务程序

3）中断程序设计需考虑的问题

中断程序设计时需要考虑的问题有以下几个方面。

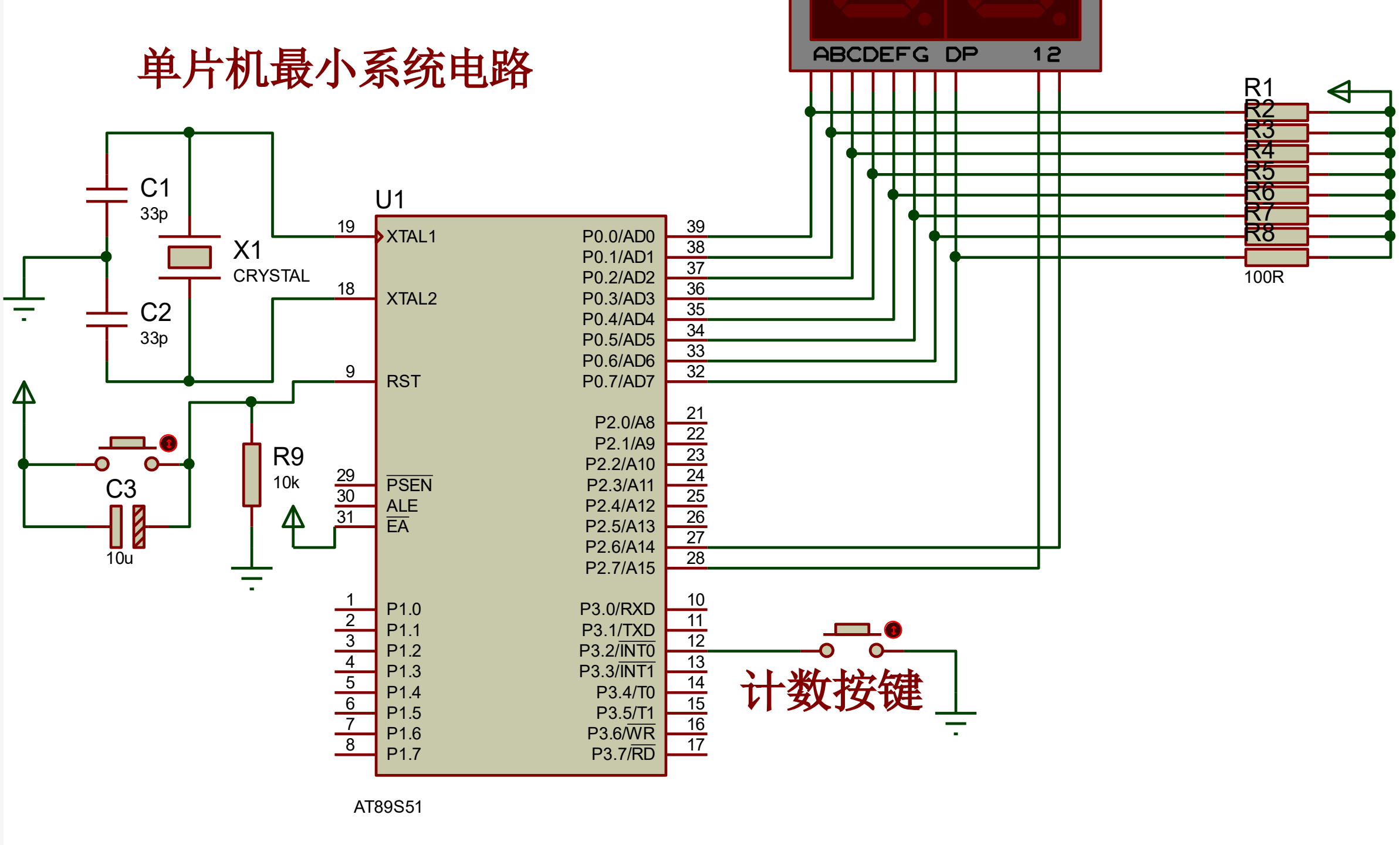
① 设置中断允许控制寄存器 IE，允许相应中断源中断。

② 设置中断优先级寄存器 IP，选择、分配所使用中断源的优先级。

③ 若是外部中断源，还要设置中断请求触发方式 IT1 或 IT0，决定采用的是边沿触发方式还是电平触发方式。

④ 编写中断服务程序，处理中断请求。前 3 条一般放在初始化主程序中。

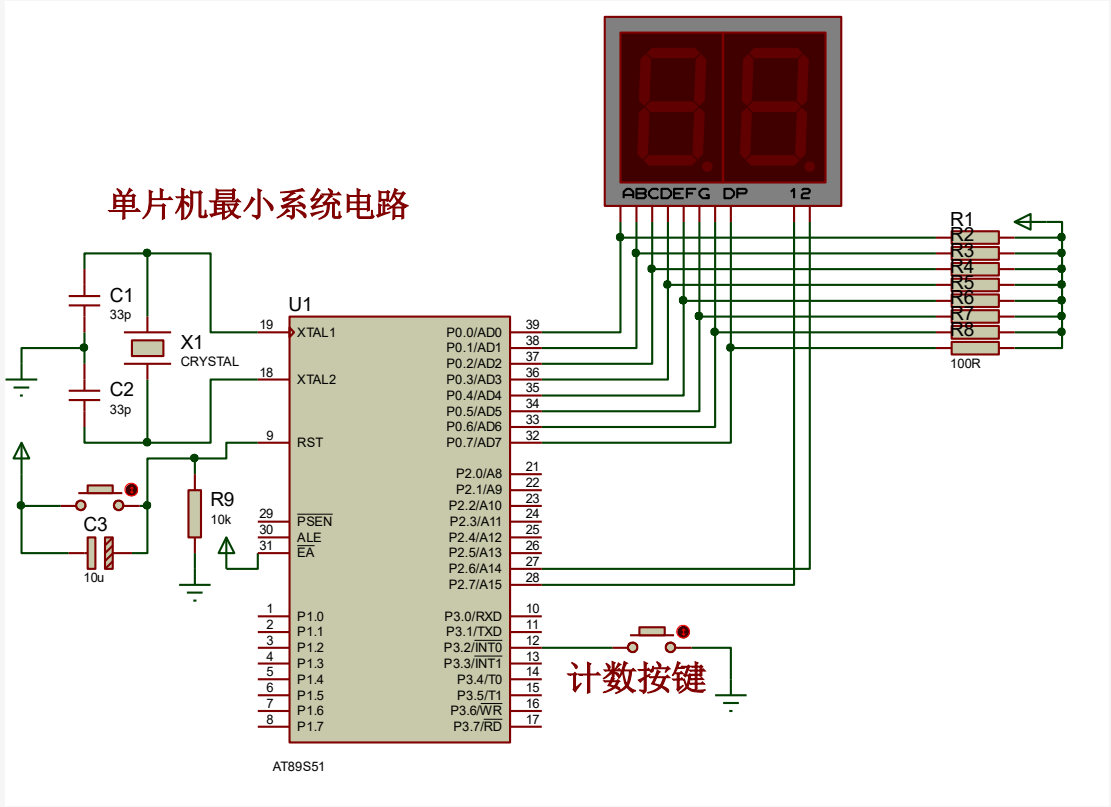
单片机最小系统电路


ABCDEFG DP 12
R1
R2
R3
R4
R5
R6
R7
R8
100R
C1
33p
X1
CRYSTAL
C2
33p
U1
19 XTAL1
18 XTAL2
9 RST
29 PSEN
30 ALE
31 EA
1 P1.0
2 P1.1
3 P1.2
4 P1.3
5 P1.4
6 P1.5
7 P1.6
8 P1.7
39 P0.0/AD0
38 P0.1/AD1
37 P0.2/AD2
36 P0.3/AD3
35 P0.4/AD4
34 P0.5/AD5
33 P0.6/AD6
32 P0.7/AD7
21 P2.0/A8
22 P2.1/A9
23 P2.2/A10
24 P2.3/A11
25 P2.4/A12
26 P2.5/A13
27 P2.6/A14
28 P2.7/A15
10 P3.0/RXD
11 P3.1/TXD
12 P3.2/INT0
13 P3.3/INT1
14 P3.4/T0
15 P3.5/T1
16 P3.6/WR
17 P3.7/RD
AT89S51
C3
10u
R9
10k
计数按键

1.3 硬件设计

2. 硬件电路

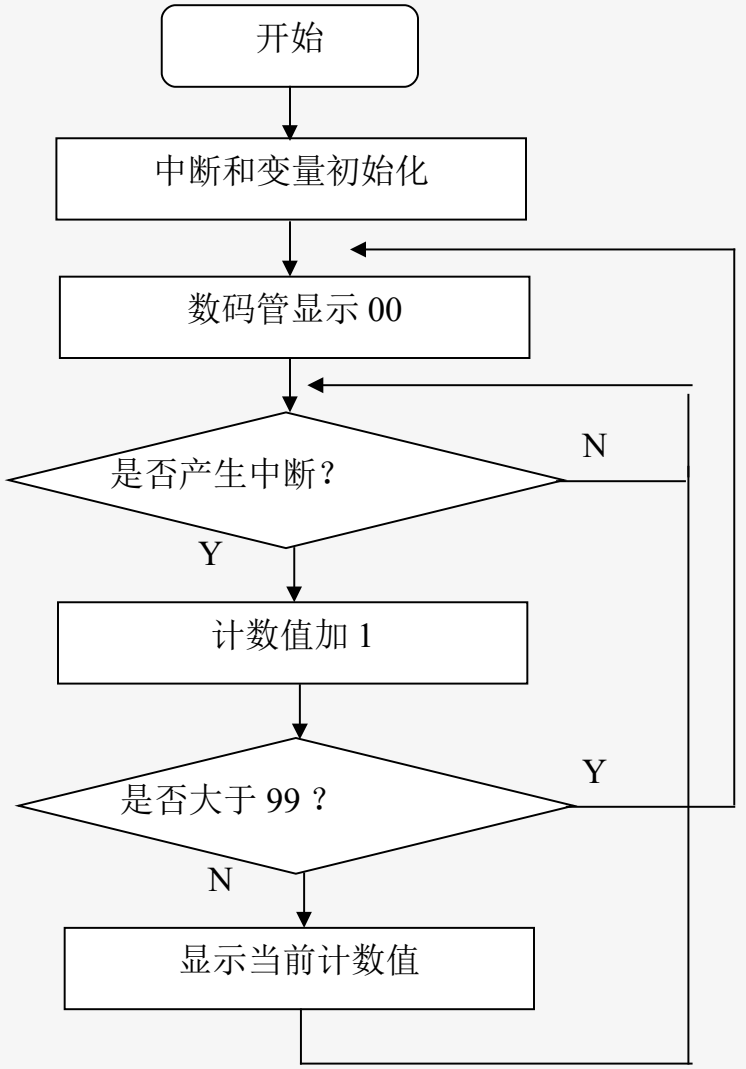
手动计数器系统仍然选用 AT89C51 单片机作为控制核心， 1 个两位一体的共阳数码管作为输出显示端。 AT89C51 的 P0 口接数码管的段码控制，其中 P0.0 ～ P0.6 分别连接数码管的 A ～ G 引脚， P0.7 连接 DP 端，低电平有效。 P2 口接数码管位码选通部分， P2.7 口控制第 1 个数码管，一直到 P2.6 口控制第 2 个数码管，高电平有效。要使 2 位数码管显示实现从 0 到 99 的动态加 1 计数，实际上就是通过 P2 口输出控制信号轮流选通数码管，共阳型数码管公共端为高电平方可选通，因此要求 P2 口由 P2.7 到 P2.0 依次输出高电平，然后在数码管段码控制端口 P0 按照一定规律送出要显示的数字 0 ～ 9 。由于 P0 口带负载能力较小，因此仿真电路中 P0 接入一排上拉电阻。为了能够获取外部按键的按下次数，外接一个按键，按键的一端连接到单片机 P3.2 ， 另一端接到地。



1.3 硬件设计

3. 软件设计

系统上电时，显示当前计数值，当有按键按下时，向单片机发出中断请求，单片机响应中断后执行中断服务，在中断服务程序中对计数值加 1，然后再返回到平时显示的状态。
程序设计时，主程序的任务主要有两个：一是执行中断初始化程序设置，二是使用数码管动态显示当前的计数值。
中断服务程序的工作就是获取外部按键的脉冲，每按下一次就把计数值加 1。
系统上电时，首先对中断寄存器初始化并初始化相关变量，数码管显示 00，然后等待按键中断，如果没有中断请求继续等待中断，如果有中断请求，就对当前计数值加 1，接着判断是否大于 99，如果不大于 99，就显示当前新的计数值，然后返回到等待中断状态，如果大于 99，就将计数值赋值 0，数码管并显示 00。如此循环。系统工作的流程图所示。



1.3 硬件设计

3. 软件设计

数码管采用的是动态显示的方法，要使 2 位数码管显示实现从 0 到 99 的动态加 1 计数，实际上就是通过 P2 口输出控制信号轮流选通数码管，共阳型数码管公共端为高电平方可选通，因此要求 P2 口由 P2.7 到 P2.0 依次输出高电平，然后在数码管段码控制端口 P0 按照一定规律送出要显示的数字 0 ～ 9 。之前采用的是对 P2 端口逐个扫描的方式，为了能够节省更多的单片机引脚，这里介绍一种方法，就是对单个引脚追个扫描。具休的方法是：先打开第 1 位数码管，送显示的内容，延时一定时间，再关闭第 1 位数码管；接着打开第 2 位数码管，送显示的内容，延时一定时间，再关闭第 2 位数码管，如此循环。具体程序如下：

```
com1 =  1;
P0 = 0xc0;
delay05ms();
com1 = 0;
com2 =  1;
P0 = 0xc0;
delay05ms();
com2 = 0;
```

1.3 硬件设计

3. 软件设计

在使用中断时，先对中断进行初始化，主要是对中断类型、中断触发的方式等进行设置，具体如下：

```
EX0 = 1；
IT0 = 1；
EA = 1；
```

初始化后需要写中断服务程序，具体程序如下：

```
void INT_0() interrupt 0
    {
     while(INT0==0);
      counter++;
  if(counter==100) counter = 0;
    }
```

总体程序见教材或电子材料

```
#include "reg51.h"
#define uchar unsigned char
uchar display_code[]={0xC0,0
xF9,0xA4,0xB0,0x99,0x92,0x8
2,0xF8,0x80,0x90,0x88,0x83,0
xC6,0xA1,0x86,0x8E};

sbit com1 = P2^7;
sbit com2 = P2^6;

uchar counter = 0;

void delay(void)
{
 uchar i;
 for(i=250;i>0;i--);
}
```

```
void display()
{
  com1 = 1;
            P0= display_co
de[counter/10];
            delay();
            com1 = 0;

            com2= 1;
            P0= display_co
de[counter%10];
            delay();
            com2 = 0;
}
```

```
void main(void)
{
  uchar i = 0;
            EX0 = 1;
            IT0 = 1;
            EA = 1;
            while(1)
            {
            for(i=0;i<250;i+
+)
                        {

            display();
                        }
            }
}
void INT_0() interrupt 0
{

            while(INT0==0)
;
            counter++;
            if(counter==10
0) counter = 0;
```

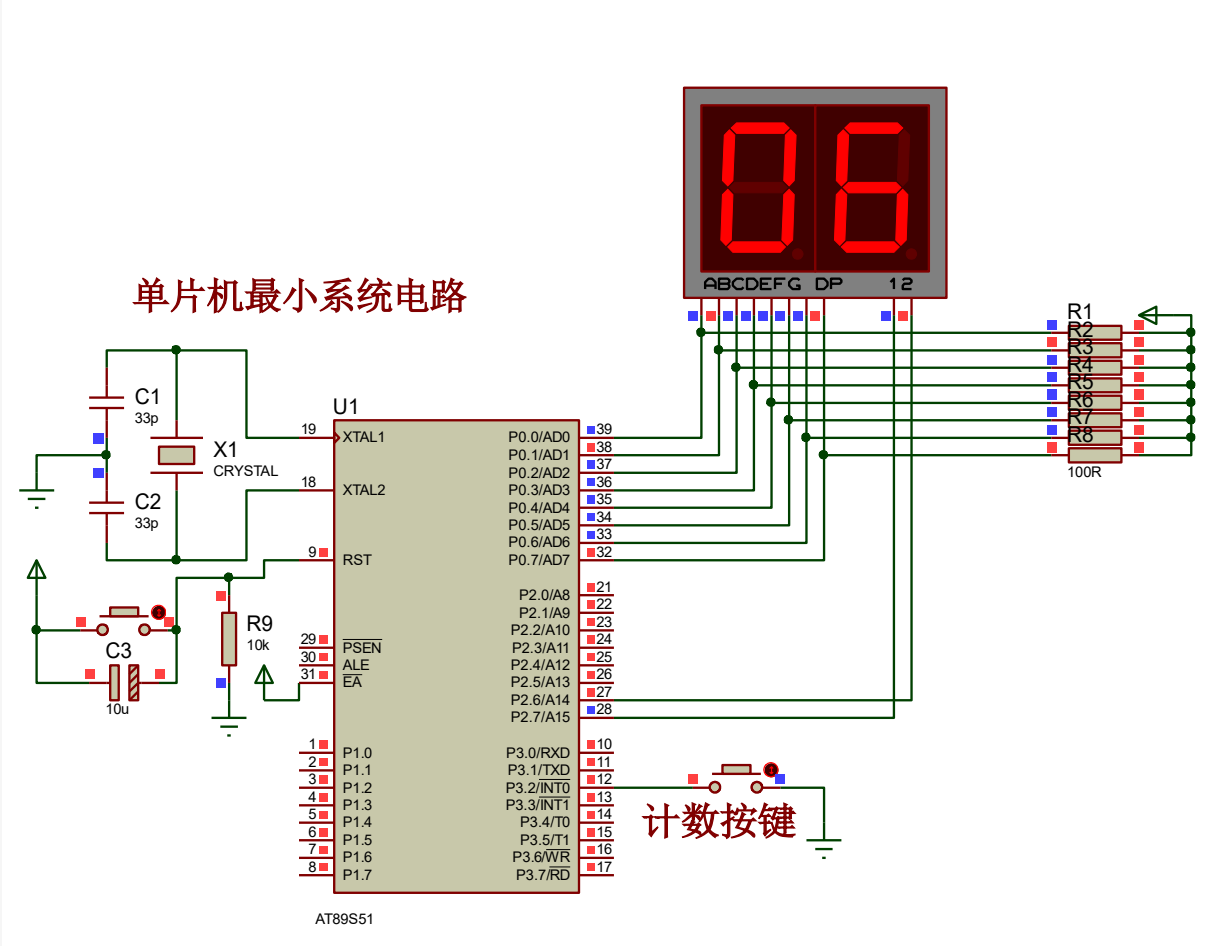
1.5 软件仿真

1. 使用 Proteus 软件，绘制如图 10-3 的硬件电路图，并保存到指定的位置。

2. 使用 KeilC51 建立一个 conuter 工程项目，建立一个 .c 文件，在编辑区中输入上述的源程序，并以“ conuter.c” 为文件名存放手动计数器文件夹中；进行编译，排除所有程序故障，直到无错误为止，目标代码文件“ conuter.hex” 。

3. 使用 Proteus 打开绘制好的按键电路设计图，双击电路图中 AT89S51 单片机，把编译好的“ conuter.hex” 文件下载到单片机中。点击模拟调试按钮，就可以看到先是数码管显示 00 ，然后每按一次按键，计数值加 1 ，计数从 0 开始，当计数到 99 后，再按一次按键，计数值变为 0 ，如此循环。

1.5 软件仿真


单片机最小系统电路
ABCDEFG DP
12
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89S51
C1
33p
C2
33p
X1
CRYSTAL
C3
10u
R9
10k
R1
R2
R3
R4
R5
R6
R7
R8
100R
计数按键

1.6 项目总结

本项目实现了手动计数器设计与仿真，介绍了单片机中断系统基本概念、中断系统的结构、中断源、以及中断相关的寄存器、中断过程等，并阐述了单片机外部中断系统编程控制的方法。

项目提升

1、单片机中断的步骤是什么？

2、简述 MSC-51 单片机中断的响应处理过程。

3、单片机的中断源共有几个？分别是什么？

4、 MSC-51 单片机外部中断有几种触发方式，分别是什么？

5、单片机的中断系统中有哪几个寄存器？试简述其作用是什么？