

模块五 单片机的中断系统

【教学聚焦】

知识目标：

- 1、了解单片机的中断系统的结构
- 2、掌握单片机的中断处理过程
- 3、掌握应用中断方式进行程序设计方法

技能目标：

- 1 能够熟悉单片机的中断系统
- 2、能够掌握输入/输出的控制方式
- 3、能在程序设计中应用中断方式

【课时建议】6 课时

教学重点：单片机的中断系统

教学难点：应用中断方式进行程序设计

项目 5.1 中断系统

中断技术是计算机中的重要技术之一，它既和硬件有关，也和软件有关。正因为有了中断，才使得计算机的工作更灵活，效率更高。本节将介绍中断的概念，并以 80C51 系列单片机的中断系统为例介绍中断的处理过程及应用。

5.1.1 中断系统的基本概念

1. 中断的概念

中断是通过硬件来改变 CPU 运行方向的一种技术，它既和硬件有关，也和软件有关。计算机在执行程序的过程中，由于单片机内部或外部的某种原因，有必要尽快终止当前程序的执行，而去执行相应的处理程序，待处理结束后，再回来继续执行被终止了的源程序。这种程序在执行过程中由于外界的原因而被中间打断的情况称为“中断”。

中断之后所执行的处理程序，通常称为“中断服务”或“中断处理子程序”，原来运行的程序称为“主程序”。主程序被断开的位置（地址）称为“断点”。引起中断的原因或能发出中断申请的来源，称为“中断源”。中断源要求服务的请求称为“中断请求（或申请）”。

调用中断服务程序的过程类似于程序设计中的调用子程序，其主要区别在于调用子程序指令在程序中是事先安排好的；而何时调用中断服务程序事先却无法确知。因为中断的发生是由外部因素决定的，程序中无法事先安排调用指令，因而调用中断服务程序的过程是由硬件自动完成的。

2. 引进中断技术的优点

计算机引进中断技术之后，主要具有如下优点：

（1）分时操作

在计算机与外部设备交换信息时，存在着高速 CPU 和低速外设（如打印机等）之间的矛盾。若采用软件查询方式，则不但占用了 CPU 操作时间，而且相应速度慢。中断功能解决了高速 CPU 与低速外设之间的矛盾。此时，CPU 在启动外设工作后，继续执行主程序，同时外设也在工作。每当外设做完一件事，就发出中断申请，请求 CPU 中断它正在执行的程序，转去执行中断服务程序（一般是处理输入/输出数据）。中断处理完成后，CPU 恢复执行主程序，外设仍然继续工作。这样，CPU 可以命令多个外设（如键盘、打印机等）同时工作，从而大大提高了 CPU 的工作效率。

（2）实时处理

在实施控制中，现场的各个参数、信息时随时间和现场情况不断变化的。有了中断功能，外界的这些变化量可根据要求随时向 CPU 发出中断请求，要求 CPU 及时处理，CPU 就可以马上响应（若中断响应条件满足）并加以处理。这样的及时处理在查询方式下是做不到的，从而大大缩短了 CPU 的等待时间。

(3) 故障处理

计算机在运行过程中，难免会出现一些无法预料的故障，如存储出错、运算溢出和电源突跳等。有了中断功能，计算机就能自行处理，而不必停机。

3. 中断源

发出中断请求的来源一般统称为“中断源”。中断源有多种，最常见的有以下4种。

(1) 外部设备中断源

计算机的输入/输出设备，如键盘、磁盘驱动器、打印机等，可通过接口电路向CPU申请中断。

(2) 故障源

故障源是产生故障信息的来源。它作为中断源，使得CPU能够以中断方式对已发生的故障及时进行处理。

计算机故障源有内部和外部之分。内部中断源一般是指执行指令时产生的错误情况，如除法中除数为零等，通常把这种中断源称为“内部软件中断”（注意：目前多数80C51系列单片机没有内部软件中断功能）；外部故障源主要有电源掉电等情况，在电源掉电时可以介入备用的电池供电电路，以保存存储器中的信息。当电压因掉电而降到一定值时，即发出中断申请，由计算机的中断系统自动响应，并进行相应处理。

(3) 控制对象中断源

计算机作实时控制时，被控对象常常用作中断源。例如电压、电流、温度等超过其上限或下限时，以及继电器、开关闭合断开时都可以作为中断源向CPU申请中断。

(4) 定时/技术脉冲中断源

定时/计数脉冲中断源也有内部和外部之分。内部定时中断是由单片机内部的定时/计数器溢出而自动产生的；外部计数中断是由外部脉冲通过CPU的中断请求输入线或定时/计数器的输入线而产生的。

要求每个中断源所发出的中断请求信号符合CPU响应中断的条件，例如电平高/低、持续时间、脉冲幅度等。

5.1.2 中断系统的结构

中断过程是在硬件基础上再配以响应的软件而实现的。不同计算机的硬件结构和软件指令是不完全相同的，因而中断系统结构一般是不相同的。但同一系列的单片机即使型号不同，中断系统的基本结构也是类似的，只是中断源个数不完全一样。

1. 中断系统的结构

51单片机中断系统主要由几个与中断有关的特殊功能寄存器、中断入口、顺序查询逻辑电路等组成。中断系统框图如图5.1所示。51单片机有5个中断源，可提供2个中断优先级，即可实现二级中断嵌套。与中断有关的特殊功能寄存器有4个，分别为中断源寄存器（即专用寄存器TCON、SCON的相关位）、中断允许控制寄存器IE和中断优先级控制寄存器IP。5个中断源的排列顺序由中断优先级控制寄存器IP和顺序查询逻辑电路共同决定。5个中断源对应5个固定的中断入口地址。

下面分别对51单片机的中断源及专用寄存器等介绍。

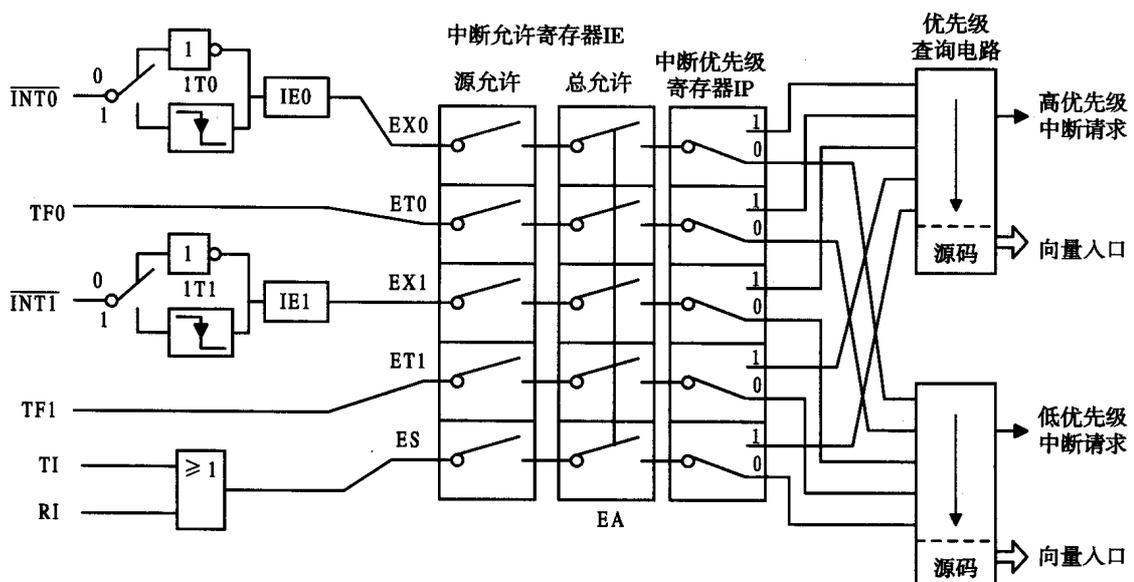


图 5.1 51 单片机的中断系统结构图

2. 中断源及入口

(1) 中断源

51 的中断源可以分为 3 类：即外部中断、定时中断和串行口中断。从图 5.1 所示的系统结构可知，51 单片机有 5 个中断请求元，分别为：2 个外部输入中断源 INT0 (P3.2) 和 INT1 (P3.3)，2 个片内定时器 T0 和 T1 的溢出中断源 TF0 (TCON.5) 和 TF1 (TCON.7)。下面分类介绍。

1 外部中断类

外部中断是由外部原因引起的，包括外部中断 0 和外部中断 1。这 2 个中断请求信号分别通过 2 个固定引脚即 INT0 (P3.2) 脚和外部中断 1 INT1 (P3.3) 脚输入。

外部中断请求信号有 2 种信号输入方式，即电平方式和脉冲方式。在电平方式下为低电平有效，即在 INT0 (P3.2) 脚或外部中断 1 INT1 (P3.3) 脚出现有效低电平时，外部中断标志下降沿时，外部中断标志 INT0 或 INT1 就置为 1。注意：在脉冲方式下，中断请求信号的高、低电平状态都应该至少维持 1 个机器周期。

中断请求信号是低电平有效还是下降沿有效，须通过 TCON 寄存器中的 IT0 (TCON.0) 或 IT1 (TCON.2) 位来决定。一旦输入信号有效，则向申请中断，并使相应的中断标志位 IE0 或 IE1 置 1。

② 定时中断类

定时中断是为满足定时或计数溢出处理的需要而设置的。

定时方式的中断请求是由单片机内部发生的，输入脉冲是内部产生的周期固定的脉冲信号（1 个机器周期），无需在芯片外部设置输入端。

计数方式的中断请求是由单片机内部发生的，脉冲信号由 T0 (P3.4) 或 T1 (P3.5) 引脚输入，脉冲下降沿为计数有效信号。这种脉冲周期是不固定的。

当定时/计数器中的计数值发生溢出时，表明定时时间或计数值已到。这时以计数溢出信号作为中断请求使溢出标志位置 1，即 T0 中断请求标志 TF0=1，或 T1 中断请求标志 TF1=1。如果允许中断，则请求中断处理。

③ 串行口中断类

串行口中断是为满足串行数据的传送需要而设置的。每当串行口由 TXD (P3.1) 端发送串行口中断请求标志 RI 或 TI 置 1，并请求中断。

由图 5.1 可以看出，当这些中断源的中断标志为 1 时，并不一定能引起中断，而是还需要经过 IE 寄存器的控制，才能进入中断请求。

(2) 中断入口

当 CPU 响应某中断源的中断申请之后，CPU 将此中断源的入口地址装入 PC，中断服务程序即从此地址开始执行，因而将此地址称为“中断入口”，亦称为“中断矢量”。在 AT89C51 单片机中，各中断源以及与之对应的入口地址（由单片机硬件电路决定）分配如表 5.1 所列。

所有的 80C51 系列单片机都有上述 5 个中断源。89 系列单片机也下例外，有些型号与其完全相同，如 Philips 公司的 P89C51；有些则增加了新的中断源，如 AT89S52 增加了定时器 T2 中断源，入口地址为 002BH；还有一些型号的中断源多达 9 个，其入口地址按 8 字节 1 个中断源顺序往下排，可以表达为：入口地址=8n+3,n 为中断自然优先级。

表 5.1 各中断源及入口地址的对应关系

中断源	入口地址
外部中断 0	0003H
定时器 T0 中断	000BH

外部中断 1	0013H
定时器 T1 中断	001BH
串行口中断	0023H

5.1.3 中断控制

中断功能虽然是硬件和软件结合的产物，但用户不必了解中断硬件电路和发生过程。对于用户来说，重点是怎样通过软件管理和应用中断功能。为此，首先应该掌握与中断控制和管理有关的几个寄存器。下面分别对其进行介绍。

(1) 中断允许控制寄存器 IE

专用寄存器 IE 是 8051 单片机中的中断允许寄存器，由它控制对中断的开放或关闭。通过向 IE 写入中断控制字，可以实现对中断的二级控制。这里所说的“二级”是指控制字中有一个中断总允许位 EA。当 EA 为 0 时，CPU 将屏蔽所有的中断申请；而当 EA 为 1 时，虽然 CPU 已经开放中断，但还须设置相应中断源的控制位，才可确定允许哪个中断源中断。

IE 各位的格式如下：

IE地址	AFH	ADH	ACH	ABH	AAH	A9H	A8H	
(A8H)	EA	—	ET2	ES	ET1	EX1	ET0	EX0

其各位名称及作用如下：

EA(IE.7)——中断允许总控制位。

当 EA=1 时，CPU 开放中断，每个中断源是被允许还是被禁止，分别由其各自的允许位决定。

当 EA=0 时，CPU 屏蔽所有的中断申请，称为“关中断”。

(IE.6)——未使用，默认为 0。

ET2(IE.5)——T2 中断允许位(仅 AT89S52/C52 或类似型号单片机有)。

当 ET2=1 时，允许 T2 中断；

当 ET2=0 时，禁止 T2 中断。

ES(IE.4)——串行口中断控制位。

当 ES=1 时，允许串行口中断；

当 ES=0 时，禁止串行口中断。

ET1(IE.3)——T1 中断控制位。

当 ET1=1 时，允许 T1 中断；

当 ET1=0 时，禁止 T1 中断。

EX1(IE.2)——外部中断 1 控制位。

当 EX1=1 时，允许外部中断 1 中断；

当 EX1=0 时，禁止外部中断 1 中断。

EX0(IE.1)——T0 中断控制位。

当 EX0=1 时，允许 T0 中断；

当 EX0=0 时，禁止 T0 中断。

EX1(IE.0)——外部中断 0 控制位。

当 EX0=1 时，允许外部中断 0 中断；

当 EX0=0 时，禁止外部中断 0 中断。

51 单片机复位后，IE 中各中断允许位均被清 0，即禁止所有中断。

例如，假定要开放外中断 1 和 T1 的溢出中断，屏蔽其它中断，则对应的中断允许控制寄存器内容应为 10001100B，即中断允许控制字为 8CH。

使用字节操作，可用一条指令 MOV IE, #8CH 完成。

使用位操作指令，则需三条指令 SETB EX1；

SETB ET1； SETB EA 实现。

(2) 中断请求标志寄存器

当有中断源发出请求时，由硬件将相应的中断标志位置 1。在中断请求被响应前、相应中断标志位被锁存在特殊功能寄存器 TCON 或 SCON 中。

① 定时器控制寄存器 TCON

TCON 为定时器 T0 和 T1 的控制寄存器，同时也锁存 T0 和 T1 的溢出中断标志及外部中断 INT0 和 INT1 的中断标志等。

TCON 中与中断有关的各位如下：

TCON	8FH		8DH		8BH	8AH	89H	
(88H)	TF1	—	TF0	—	IE1	IT1	IE0	IT0

其各位名称及作用如下：

TF1(TCON.7) —— T1 溢出中断标志

当 T1 开始工作，并且计数值产生溢出时，由硬件使 TF1=1，在中断工作方式下向 CPU 请求中断。此标志一直保持到 CPU 响应中断后，才由硬件自动清 0；也可用软件查询该标志，并由软件清 0。

如果 T1 不工作，或者在工作但没有产生溢出，则 TF1=0

TF0(TCON.5) —— T0 溢出中断标志。

其操作功能类似于 TF1。

IE1(TCON.3) —— INT1 外部中断 1 标志。

当硬件使 IE1=1 时，表明外部中断 1 向 CPU 申请中断。

当 IE1=0 时，表明外部中断 1 没有向 CPU 申请中断。

IT1(TCON.2) —— 外部中断 1 触发方式控制位。

当 IT1=0 时，外部中断 1 设置为电平触发方式。在这种方式下，CPU 在每个机器周期的 S5P2 期间对 INT1 (P3.3) 引脚采样。若采样为低电平，则认为有中断申请，随即使 IE1 标志置 1；若为高电平，则认为无中断申请或中断申请已撤除，随即清除 IE1 标志。在电平触发方式中，CPU 响应中断后不能自动清除 IE1 标志，也不能由软件清除 IE1 标志，故在中断返回前，必须撤销 INT1 引脚上的低电平，否则将会引起再次中断而出错。

当 IT1=1 时，外部中断 1 设置为边沿触发方式。CPU 在每个机器周期的 S5P2 期间采样引脚，若在连续 2 个机器周期采样到先高电平后低电平，则使 IE1 标志置 1，此标志一直保持到 CPU 响应中断时，才由硬件自动清除。在边沿触发方式中，为保证 CPU 在 2 个机器周期内检测到先高后低的负跳变，则输入高/低电平的持续时间起码要保持 12 个时钟周期。

IE0(TCON.1) —— INT0 外部中断 0 标志位。

其操作功能与 IE1 类似。

IT0(TCON.0) —— 外部中断 0 触发方式控制位。

其操作功能与 IT1 类似。

TCON.6 和 TCON.4 —— 中断中未使用。

② 串行口控制寄存器 SCON

SCON 是串行口控制寄存器，其低 2 位 TI 和 RI 锁存串行口的接收中断和发送中断标志。

SCON 中与中断有关的各位如下：

SCON							99H	98H
(98H)	—	—	—	—	—	—	TI	RI

其各位名称及作用如下：

TI(SCON.1)——串行发送中断标志。

当 TI=1 时，说明 CPU 将 1 字节数据写入发送缓冲器 SBUF。并且已发送完 1 个串行帧，此时，硬件使 TI 置 1。在中断工作方式下，可以向 CPU 申请中断，在中断和查询工作方式下都不能自动清除 TI，必须由软件清除标志。

当 TI=0 时，说明没有进行串行发送，或者串行发送未完成。

RI(SCON.0)——串行接收中断标志。

当 RI=1 时，在串行口允许接收后，每接收完 1 个串行帧，硬件使 RI 置 1。同样，在中断和查询工作方式下都不会自动清除 RI，必须由软件清除标志。

当 RI=0 时，说明没有进行串行接收，或者串行接收来完成。

注意：SCON 的高 6 位在中断中不用，所以在此不做介绍。

(3) 中断优先级寄存器 IP

80C51 单片机中断优先级的设定由专用寄存器 IP 统一管理。

它具有 2 个中断优先级，由软件设置每个中断源为高优先级中断或低优先级中断，可实现二级中断嵌套。

高优先级中断源可中断正在执行的低优先级中断服务程序，除非在执行低优先级中断服务程序时设置了 CPU 关中断或禁止某些高优先级中断源的中断。同级或低优先级的中断源不能中断正在执行的中断服务程序。为此，在 51 中断系统中，内部有 2 个（用户不能访问的）优先级状态触发器，它们分别指示 CPU 是否在执行高优先级或低优先级中断服务程序，从而决定是否屏蔽所有的中断申请或同一级的其他中断申请。

专用寄存器 IP 为中断优先级寄存器，用于选择各中断源优先级，用户可用软件设定。其各位格式如下：

IP	BDH		BCH	BBH	BAH	B9H	B8H	
(B8H)	—	—	PT2	PS	PT1	PX1	PT0	PX0

其各位名称及作用如下：

PT2(IP.5)——T2 中断优先级选择位（仅 AT89S52、C52 或类似型号单片有）。

当 PT2=1 时，设置定时器 T2 为高优先级中断；

当 PT2=0 时，设置定时器 T2 为低优先级中断。

PS(IP.4)——串行口中断优先级选择位。

当 PS=1 时，设定串行口为高优先级中断；

当 PS=0 时，设定串行口为低优先级中断。

PT1 (IP.3)——T1 中断优先级选择位。

当 PT1=1 时，设定定时器 T1 为高优先级中断；

当 PT1=0 时，设定定时器 T1 为低优先级中断。

PX1 (IP.2)——外部中断 1 中断优先级选择位。

当 PX1=1 时，设定外部中断 1 为高优先级中断；

当 PX1=0 时，设定外部中断 1 为低优先级中断。

PT0(IP.1)——T0 中断优先级选择位。

当 PT0=1 时，设定定时器 T0 为高优先级中断；

当 PT0=0 时，设定定时器 T0 为低优先级中断。

PX0(IP.0)——外部中断 0 中断优先级选择位。

当 PX0=1 时，设定外部中断 0 为高优先级中断；

当 PX0=0 时，设定外部中断 0 为低优先级中断。

当系统复位后，IP 全部清 0，将所有中断源设置为低优先级中断。

如果几个相同优先级的中断源，同时向 CPU 申请中断，CPU 通过内部硬件查询逻辑按自然优先级顺序确定响应哪个中断请求。其自然优先级由硬件形成，排列如表 5.2 所列。

表 5.2 各中断源及其自然优先级

序号	中断源	自然优先级
1	外部中断 0	最高级 ↓ 最低级
2	定时器 T0 中断	
3	外部中断 1	
4	定时器 T1 中断	
5	串行口中断	

这种排列顺序在实际应用中很方便、合理。如果重新设置了优先级，则顺序查询逻辑电路将会相应改变排队顺序。例如：如果给 IP 中设置的优先级控制字为 09H，则 PT1 和 PX0 均为高优先级中断，但当这 2 个中断源同时发出中断申请时，CPU 将首先响应自然优先级较高的 PX0 的中断申请。

例如，某软件中对寄存器 IE、IP 设置如下：

MOV IE, #8FH◆

MOV IP, #06H◆

则此时该系统中：CPU 中断允许；允许外部中断 0、外部中断 1、定时器/计数器 0、定时器/计数器 1 提出的中断申请；○◆

允许中断源的中断优先次序为：◆

定时器/计数器 0 > 外部中断 1 > 外部中断 0 > 定时器/计数器 1。

5.1.4 中断过程

中断处理过程可分为 3 个阶段，即中断响应、中断处理和中断返回。所有计算机的中断处理都有这样 3 个阶段，但不同的计算机由于中断系统的硬件结构不完全相同，因而中断响应的方式有所不同。在此仅以 51 系列单片机为例来介绍中断处理过程。

1. 中断响应

中断响应是在满足 CPU 的中断响应条件之后，CPU 对中断源中断请求的回答。在这一阶段，CPU 要完成执行中断服务以前的所有准备工作。这些准备工作包括保护断点和把程序转向中断服务程序的入口地址（通常称为“矢量地址”）。

计算机在运行时，并不是任何时刻都会去响应中断请求，而是在中断响应条件满足之后才会响应。

(1) CPU 的中断响应条件

CPU 响应中断的条件主要有以下几点：

- ◆ 有中断源发出中断请求；
- ◆ 中断总允许位 EA=1，即 CPU 允许所有中断源申请中断；
- ◆ 申请中断的中断源的中断允许位为 1，即此中断源可以向 CPU 申请中断。

以上是 CPU 响应中断的基本条件。若满足，CPU 一般会响应中断，但如果有下列任何一种情况存在，则中断响应就会受到阻断。

- ◆ CPU 正在执行一个同级或更高级的中断服务程序；
- ◆ 当前的机器周期不足正在执行指令的最后一个周期，即正在执行的指令完成前，任何中断请求都得不到响应；
- ◆ 正在执行的指令是返回 (RETI) 指令或者对专用寄存器 IE、IP 进行读 / 写的指令，此时，在执行 RETI 或者读 / 写 IE 或 IP 之后，不会马上响应中断请求。至少要再执行一条其他指令，才会响应中断。

存在上述任何一种情况，CPU 都不会马上响应中断。此时将该中断请求锁存在对应中断源的中断标志位中，然后在下一个机器周期再按顺序查询。

在每个机器周期的 S5P2 期间，CPU 对各中断源采样，并设置相应的中断标志位。CPU 在下一个机器周期 S6 期间按优先级顺序查询各中断标志，如查询到某个中断标志为 1，将在再下一个机器周期 S1 期间按优先级进行中断处理。中断查询在每个机器周期中重复执行，如果中断响应的基本条件已满足，但由于上述原因之一而未被及时响应，待封锁中断的条件撤销之后，由于中断标志还存在，故仍会响应。

(2) 中断响应过程

如果中断响应条件满足，且不存在中断阻断的情况，则 CPU 将响应中断。

在 51 单片机的中断系统中有 2 个优先级状态触发器，即“高优先级状态”触发器和“低优先级状态”触发器。这 2 个触发器是由硬件自动管理的，用户不能对其编程。当 CPU 响应中断时，它首先使优先级状态触发器置位，这样可以阻断同级或低级的中断；然后，中断系统自动把断点地址压入堆栈保护（但不保护状态寄存器 PSW 及其他寄存器内容），再由硬件执行一条长调用指令将对应的中断入口装入程序计数器 PC，使程序转向该中断入口地址，并执行中断服务程序。

2. 中断处理

中断处理程序（又称“中断服务”或“中断子程序”）从入口地址开始执行，直到返回指令 RETI 为止。这个过程称为“中断处理”。此过程主要用于处理中断源的请求，但由于中断处理程序是由随机事件引起的实时响应，从而使得它与一般的子程序存在一定差别。

在编写中断服务程序时，须注意以下几点：

- ◆ 注意保护现场和恢复现场，因为一般主程序和中断服务程序都可能会用到累加器、PSW 寄存器及其他一些寄存器。CPU 进入中断服务程序后，在用到上述寄存器时，就会破坏它原来存在寄存器中的内容，一旦中断返回，将会造成主程序的混乱，因而在进入中断服务程序后，一般要先保护现场，然后再执行中断处理程序，并在返回主程序以前恢复现场。对于要保护的内容一定要全面考虑，不能遗漏。
- ◆ 在 CPU 响应中断，使程序转向该中断入口地址后，通常不能从此地址开始运行中断服务程序，因为各入口地址之间只相隔 8 字节，一般的中断服务是容纳不下的。因此，最常用的方法是在中断入口地址单元处存放一条无条件转移指令，使程序跳转到用户安排的中断服务程序起始地址上去。这样，可使中断服务程序灵活地安排在 64KB 程序存储器的任何空间。
- ◆ 若要在执行当前中断程序时禁止更高优先级中断源产生的中断，应先用软件关闭 CPU 中断，或屏蔽更高级中断源的中断，在中断返回后再开放中断。
- ◆ 在保护现场和恢复现场时，为了不使现场数据受到破坏或者造成混乱，一般规定此时 CPU 不响应新的中断请求。这就要求在编写中断服务程序时，注意在保护现场之前要关中断，在恢复现场之后开中断。如果在中断处理时，允许有更高级的中断打断它，则在保护现场之后再开中断，恢复现场之前关中断。

3. 中断返回

中断返回是指中断服务完成后，计算机返回到断点（即原来断开的位置），继续执行原来的程序。中断返回由专门的中断返回指令 RETI 实现。

该指令的功能是将断点地址取出，送回到程序计数器 PC 中。另外，它还通知中断系统已完成中断处理，清除优先级状态触发器，并使部分中断源标志（除 TI、RI）清 0。

在中断服务程序中，要特别注意不能用 RET 指令代替 RETI 指令。

4. 中断请求的撤销

CPU 响应某中断请求后，在中断返回前应撤销该中断请求，否则会引起另一次中断。

对定时器 T0 或 T1 溢出中断，CPU 在响应中断后，就用硬件清除了有关的中断请求标志 TF0 或 TFI。即中断请求是自动撤除的，无须采取其他措施。

对于边沿触发的外部中断，CPU 在响应中断后，也是用硬件自动清除有关的中断请求标志 IE0 或 IEI。无须采取其他措施。

对于串行口中断，CPU 响应中断后，没有用硬件清除 TI、RI，故这些中断标志不能自动撤除，而要靠软件来清除相应的标志。

以上中断的撤销都较简单，只有对电平激活的外部中断，其撤销方法较复杂。因为在电平触发方式中，CPU 响应中断时不会自动清除 IE1 或 IE0 标志，所以应在响应中断后立即撤销 INT0 或 INT1 引脚上的低电平。因为 CPU 对 INT0 和 INT1 引脚的信号不能控制，所以这个问题要通过硬件，并配合软件来解决。图 5.2 所示为可行方案之一。外部中断请求信号不直接加在 INT0 或 INT1 上，而须加在 D 触发器的 CLK 中断。由于 D 端接地，当外部中断请求的正脉冲信号出现 CLK 端时，INT0 或 INT1 为低，发出中断请求。用 P1.0 接在触发器的 S 端作为应答线，当 CPU 响应中断后可用如下 2 条指令：

```
CLR P1.0
SETB P1.0
```

执行第 1 条指令使 P1.0 输出为 0，其持续时间为 2 个机器周期，足以使 D 触发器置位，从而撤销中断请求。执行第 2 条指令使 P1.0 变为 1，否则 D 触发器的 S 端始终有效，INT0 端始终为 1，无法再次申请中断。

除此以外，还可以采用其他方法。

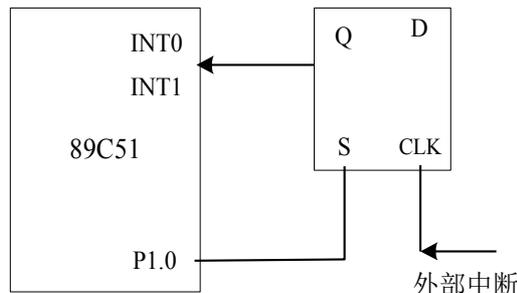


图 5.2 撤销电平激活的中断方案之一

5. 中断响应时间

由上述可知，CPU 不是在任何情况下都对中断请求予以响应的。此外，不同的情况对中断响应的的时间也是不同的。下面以外部中断为例，说明中断响应的的时间。

在每个机器周期的 S5P2 期间，INT0、INT1 端的电平分别被锁存到 TCON 的 IE0、IE1 位，CPU 在下一个机器周期才会查询这些值。此时如果满足中断响应条件，下一条要执行的指令将是一条硬件长调用指令 LCALL，使程序转入中断矢量入口。调用本身要用 2 个机器周期，这样从外部中断请求有效到开始执行中断服务程序的第 1 条指令，至少需要 3 个机器周期，这是最短的响应时间。

如果遇到中断受阻的情况，则中断响应时间会更长一些。例如，当一个同级或更高级的中断服务正在进行，则附加的等待时间取决于：正在进行的中断服务程序；如果正在执行的一条指令还没有进行到最后一个机器周期，附加的等待时间为 1~3 个机器周期（这是由于一条指令的量长执行时间为 4 个机器周期），如果正在执行的是 RETI 指令或者访问 IE 或 IP 的指令，则附加的等待时间在 5 个机器周期之内（完成正在执行的指令还需要 1 个机器周期，加上完成下一条指令所需的最长时间 4 个机器周期，故最长为 5 个机器周期）。

若系统中只有一个中断源，则响应时间为 3~8 个机器周期。如果有 2 个以上中断源同时申请中断，则响应时间将更长。一般情况下可不考虑响应时间，但在精确定时的场合则须考虑此问题。

6. 扩充外部中断源

51 单片机具有 2 个外部中断请求输入端 INT0 或 INT1。在实际应用中，若外部中断源超过 2 个，就需扩充外部中断源。这里介绍两种比较简单、可行的方法。

(1) 利用定时器扩展外部中断源法

51 单片机有 2 个定时器，具有 2 个内部中断标志和外部计数引脚。将定时器设置为计数方式，计数初值设定为满量程，一旦从外部计数引脚输入一个负跳变信号，计数器即加 1 产生溢出中断。把外部计数输入端 T0(P3.4)或 T1(P3.5)作扩充中断源输入。该定时器的溢出中断标志及服务程序作扩充中断源的标志及服务程序。

例如：特定定时器 T0 设定为方式 2（自动重装载常数）代替一个扩充外部中断源，TH0 和 TL0 初值为 FFH，允许 T0 中断，CPU 开放中断，初始化程序如下：

```
MOV TMOD, #06H
MOV TL0, #0FFH
MOV TH0, #0FFH
SETB TR0
SETB ET0
SETB EA
```

当连接在 T0(P3.4)引脚的外部中断请求输入线发生负跳变时，TL0 计数加 1 产生溢出，置位 TF0 标志，向 CPU 发出中断申请，同时 TH0 的内容 FFH 送到 TL0，即 TL0 恢复初值。T0 引脚每输入 1 个负跳变信号，TF0 都会置 1，且向 CPU 请求中断，这就相当于边沿触发的外中断源输入了。

(2) 中断和查询结合法

采用定时器的方法有一定的局限性，只能增加 2 路中断源。如果要扩充更多的外部中断源，可以采用中断和查询结合的方法。图 5.3 所示就是一种扩充外部中断源的实用方法。

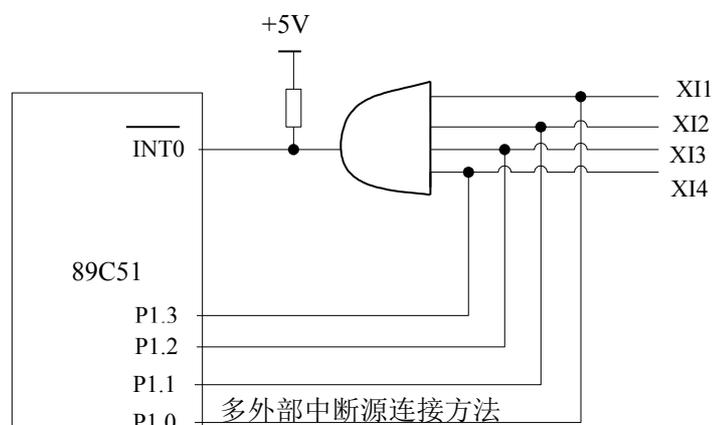


图 5.3 中，采用一个外部中断源扩充 4 个外部中断源，所有这些扩充的外部中断源都是电平触发方式（低电平有效）。当 4 个扩充中断源 XI1~XI4 中有一个或几个出现低电平时，“与”门输出为 0，使 INT1 为低电平触发中断。在外中断 1 服务程序中，由软件接入为设定的顺序（优先级）查询外部中断源哪位是高电平，然后进入该中断进行处理。

在本例中，各路输入的有效中断电平应该在 CPU 实际响应该中断源之前保持有效，并在该中断服务程序返回前取消。

INT1 的中断服务程序如下：

```
EXINT: PUSH PSW
      PUSH ACC
      JNB P1.0, AV1
      JNB P1.1, AV2
```

```

JNB P1.2, AV3
JNB P1.2, AV4
DIB: POP ACC
POP PSW
RETI
AV1: ; XI1 中断服务程序
:
LJMP DIB
AV2: ; XI2 中断服务程序
:
LJMP DIB
AV3: ; XI3 中断服务程序
:
LJMP DIB
AV4: ; XI4 中断服务程序
:
LJMP DIB

```

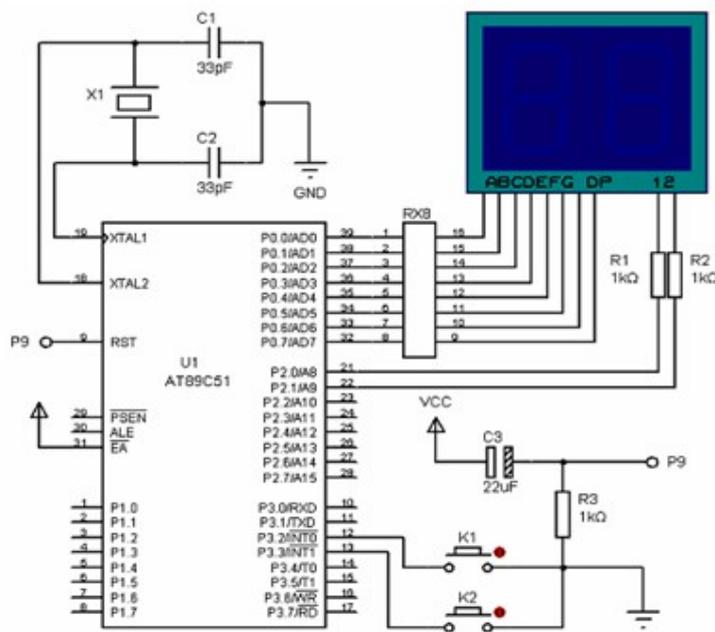
项目 5. 2 应用项目三、电子计数器应用

5.2.1 中断初始化

电子计数器在现实生活中应用非常的广泛，本项目将设计一个电子计数器，单片机的 P3.2（外部中断 0）P3.3（外部中断 1）分别连接了两个按键 K1，K2，当仅按下 K1 键时，数码管循环显示 00→01…99→00，当仅按下 K2 键时，数码管循环显示 99→98…00→99，当无键按下或两个键都按下时，数码管循环显示 00。

5.2.2 电子计数器的实现

1、电路原理图



2、元器件清单

序号	名称	数量
1	51 单片机	1 个
2	排阻尺	1 个
3	电解电容	22uF

4	瓷片电容	33pF	1 个
5	晶振 X1	CRYSTAL	1 个
6	开关		2 个
7	电阻	1K	3 个
8	数码管 共阴		2 个

3、源程序清单

```

#include <reg51.h>
unsigned b[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
void delay( void )          ; 延时子程序
{
    unsigned char i,j,k;
    for( i=20; i>0; i-- )
        for( j=2; j>0; j-- )
            for( k=250; k>0; k--);
}
void Inc_CNT ( void )      ; K1 按下显示 00→01…99→00
{
    unsigned char x, y, z;
    for( x=0; x<10; x++){
        for( y=0; y<10; y++){
            if( (P3|0xfb)==0xfb ){
                for( z=0; z<22; z++){
                    P2=0xf2;
                    P0=b[x];
                    delay( );
                    P2=0xf1;
                    P0=b[y];
                    delay( );
                }
            }
            else {
                P0 = 0x3f;
                P2 = 0x00;
                break;
            }
        }
    }
}
void Dec_CNT ( void )     ; 按下 K2 键显示 99→98…00→99
{
    unsigned char x, y, z;
    for( x=10; x>0; x--){
        for( y=10; y>0; y--){
            if( (P3|0xf7)==0xf7 ){
                for( z=0; z<22; z++){

```

```

        P2=0xf2;
        P0=b[x-1];
        delay( );
        P2=0xf1;
        P0=b[y-1];
        delay( );
    }
}
else {
    P0 = 0x3f;
    P2 = 0x00;
    break;
}
}
}
}
void main ( void )
{
    unsigned char key;
    P2=0x00;
    P0=0x3f;
    for( ; ; ){
        P3|=0xf3;
        key=P3;
        if( key==0xfb ) Inc_CNT();
        else if( key==0xf7 ) Dec_CNT();
        else{
            P2=0x00;
            P0=0x3f;
        }
    }
}

```

4、仿真运行结果图

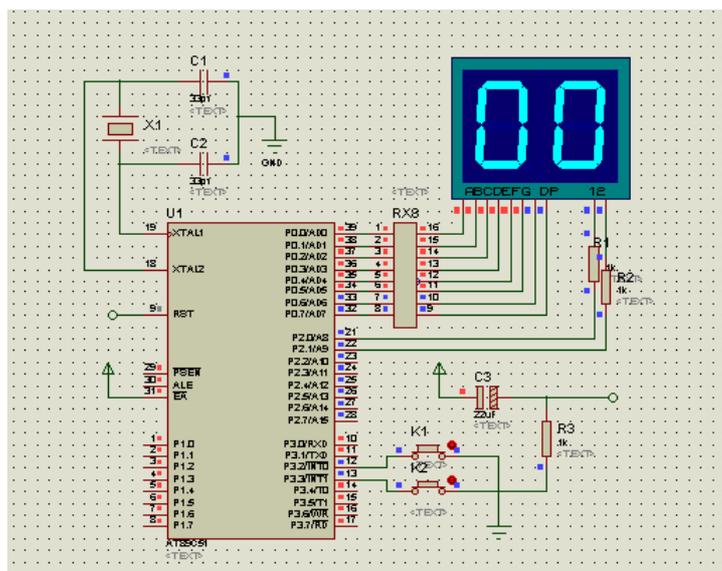


图 5.4 电子计数器显示 00

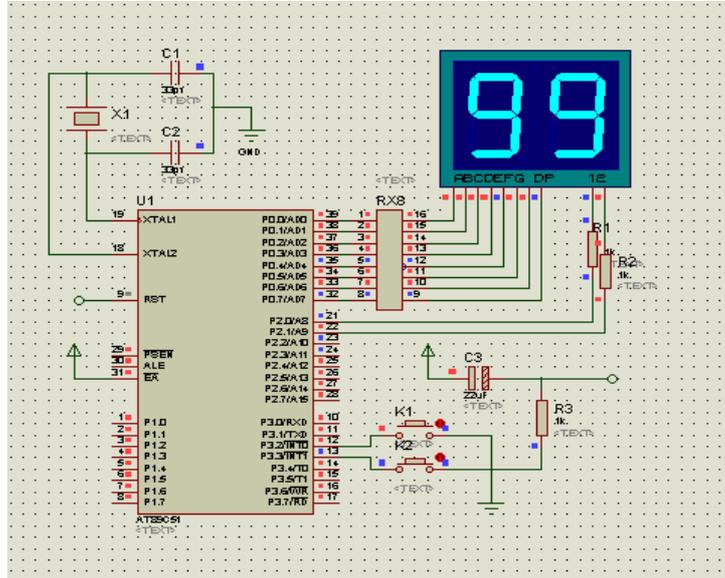


图 5.5 应用项目四、九级电压任意应用

5.3.1 单片机控制分析

(1) 单片机最小系统的结构

AT 系列单片机最小系统是以 AT 系列单片机作为核心，再加上外围电路使其组成最小系统，主要外围电路包括时钟电路、复位电路和扩展接口电路等部分，其组成结构如图 5.6。

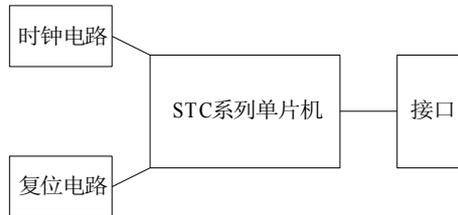


图 5.6 单片机最小系统图

我们采用 AT89C51 单片机作为控制系统，其引脚配置如 5.7 图所示。主要包括 P0、P1、P2 和 P3 口，P3 口功能最多，不仅可以读写使能输出，还可以数据传输等。

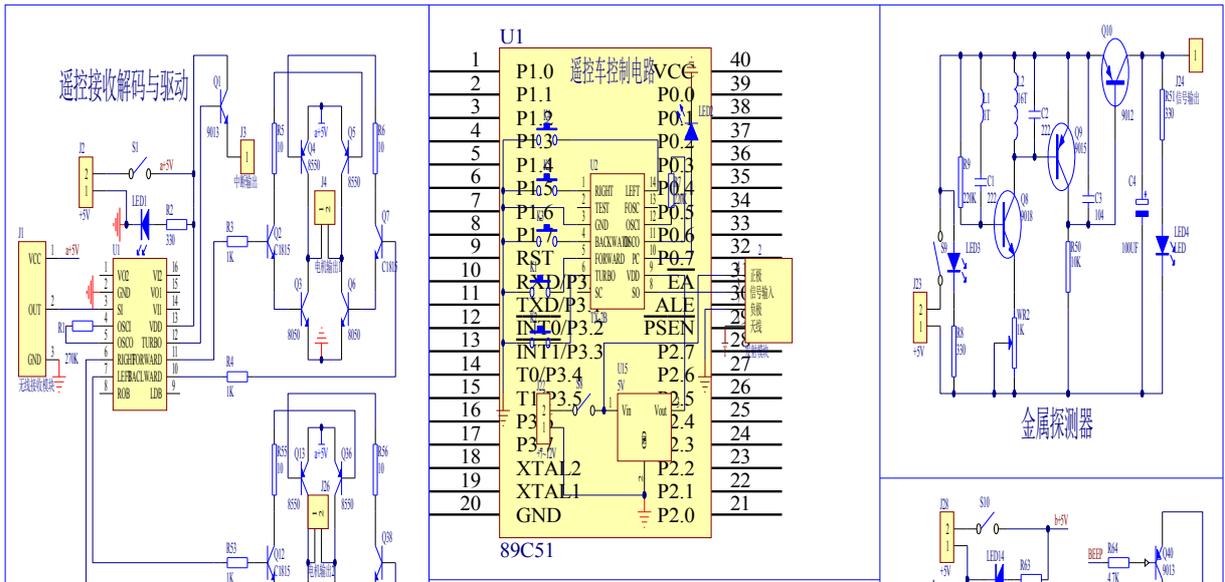


图 5.7 STC89C51 单片机引脚分布

(2) 时钟电路

在单片机系统中，一般在单片机的引脚 XTAL1 和 XTAL2 之间接一个晶振和两个电容，这样构成了单片机系统的内部振荡方式，由于单片机内部有一个高增益反相放大器，外接一个晶振构成了自激振荡器并产生震荡时钟脉冲。如图 5.8。

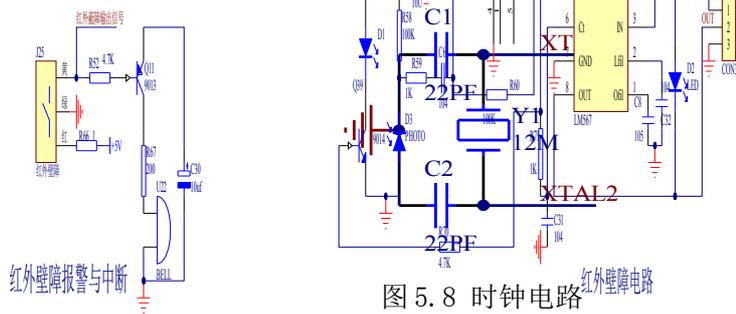


图 5.8 时钟电路

(3) 复位电路

在 STC 系列单片机系统中，一般需要一个硬件复位电路，用于手动复位和启动自动复位，接高电平有效。最简单的复位电路有一个电阻、电容和按键组成，如图 5.9 中。当单片机系统接通电源后，电容自动充电产生自动复位。如果手动复位，就是在接通电源的条件下按下复位按键 K1 就可以实现单片机复位。

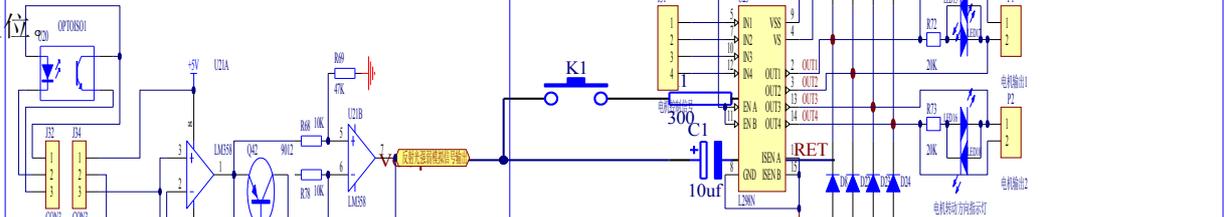


图 5.9 复位电路

5.3.2 无线电遥控车电路分析

无线电遥控小车首先具备沿着设定的路线寻走，在寻走的基础上可能有一个障碍物，所以要有避开障碍物的功能。然后还能够在路面探测金属，为了更容易控制小车，还需要一个无线遥控功能，在自动循迹的同时，无线遥控功能优先。最后多功能智能小车要有实用性，它还要能在山地上正常使用。如图 5.10。

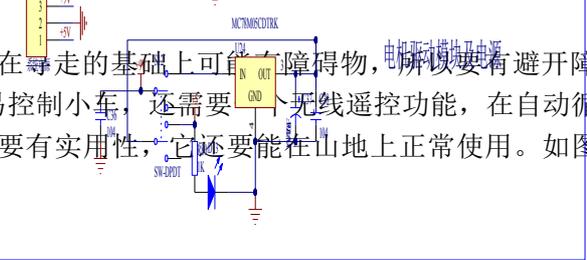


图 5.10 无线电遥控车电路图

无线电遥控车由 8 个模块构成，分别是单片机控制系统模块、单片机驱动模块及电源、红外循迹模块、红外避障模块、红外报警模块、金属探测模块、无线电发射与接收模块，及遥控车控制模块。

5.3.3 编写程序

(1) 程序设计分析

智能小车是由单片机控制的，而单片机是要依靠程序来运行的，所以程序是智能小车的主要部分。综合各功能模块的信号，由单片机采集起来，根据设定的程序让智能小车运作，也就是说智能小车所执行的都是事先在程序设定好的，所以智能小车所遇到的情况要考虑周到才能让它依照你的想法去做。

根据第二章中的智能小车设计框架，其程序设计框图如下图 5.11：

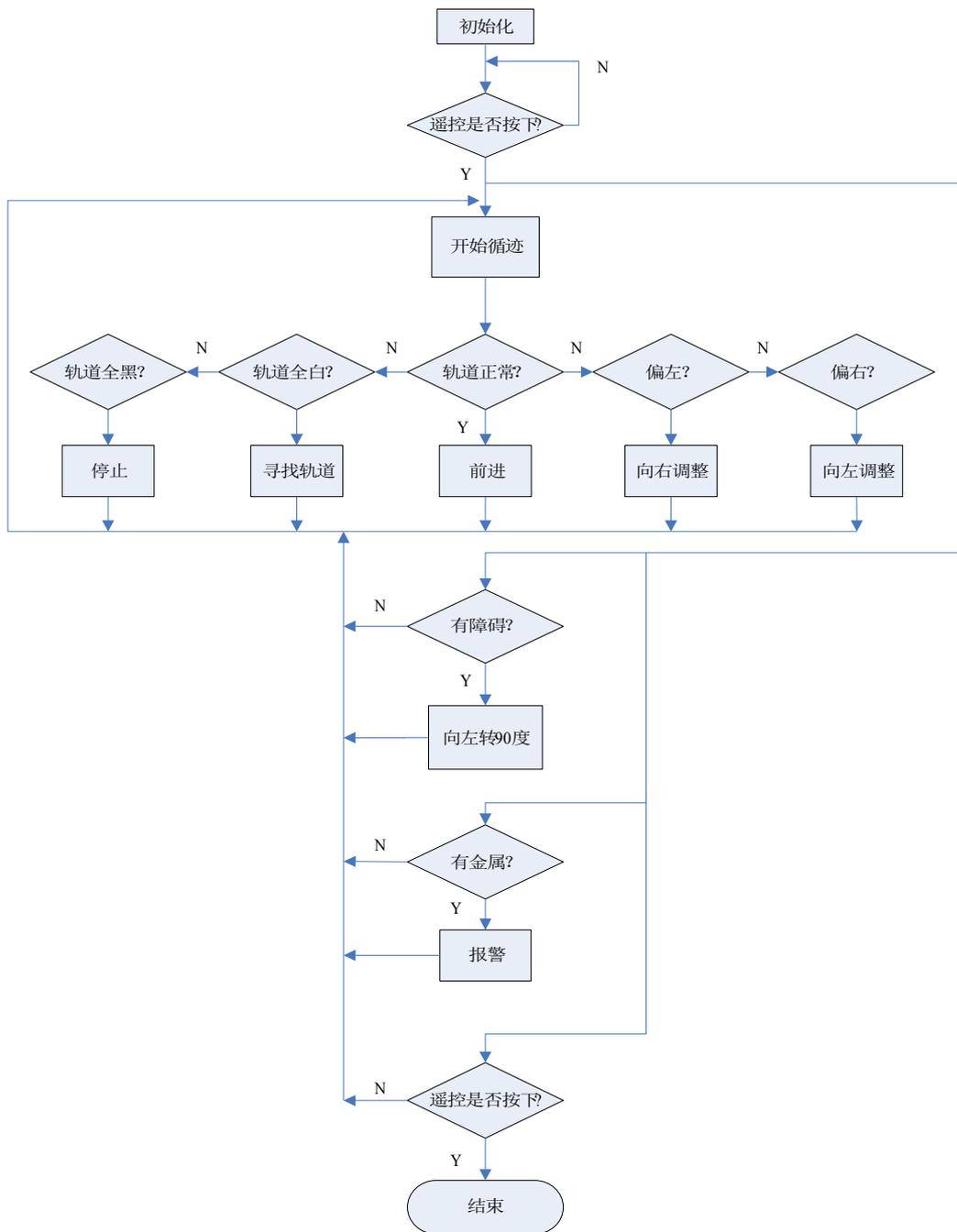


图 5.11 软件设计流程图

(2) 源程序

```

#include <reg51.h>
#define Shake 10;
void delay_1s(unsigned char );
//char mode_choose();
sbit Infrared_alarm=P1^0;
sbit Metal_detector=P1^1;
//sbit Mode=P1^2;
sbit Warning=P1^7;
sbit interrupt_0=P3^2;

bit m_mode=0,in_Alarm;
  
```

```

bit black_0,white_1,white_4;
main()
{
    P3=0x04;
    EA=1;
    EX0=1;
    IT0=1;
    Metal_detector=0;
    while(1)
    {
        if(m_mode==1)
            {
                while(1)
                {
                    P2=0x00;
                    if(Metal_detector==1)
                    {
                        P2=P2&0x0f|0x00;
                        Warning=0;
                        delay_1s(200);
                        Warning=1;
                        delay_1s(200);
                    }
                    if(m_mode==0)
                    break;
                }
            }
        while(1)
        {
            //P2=P2&0x0f|0x00;
            Warning=1;
            switch(P2&0x07)
            {
                case 0:
                    P2=P2&0x0f|0x00;
                    black_0=1;
                    break;
                case 1:
                    white_1=1;
                    P2=P2&0x0f|0x90; break;
                case 2:
                    white_1=1;
                    P2=P2&0x0f|0x90; break;
                case 3:
                    P2=P2&0x0f|0x90; break;
                case 4:

```

```

        white_4=1;
        P2=P2&0x0f|0x60; break;
    case 5:
        P2=P2&0x0f|0xA0; break;
    case 6:
        white_4=1;
        P2=P2&0x0f|0x60; break;
    case 7:
        black_0=1;
        break;
    default: break;
}
if(black_0==1&&white_1==1)
{
    while((P2&0x07)==1)
    {
        P2=P2&0x0f|0x90;
        white_4=0;
        black_0=0;
    }
}
if(black_0==1&&white_4==1)
{
    while((P2&0x07)==4)
    {
        P2=P2&0x0f|0x60;
        white_1=0;
        black_0=0;
    }
}
while(~Infrared_alarm)
{
    P2=P2&0x0f|0x90;
    //delay_1s(50);
    in_Alarm=1;
}
if(in_Alarm==1)
{
    P2=P2&0x0f|0x00;
    in_Alarm=0;
}
if(Metal_detector==1)
{
    P2=P2&0x0f|0x00;
    Warning=0;
    delay_1s(200);
}

```

```

        Warning=1;
        delay_1s(200);
    }
    if(m_mode==1)
        break;

}
}
}
void delay_1s(unsigned char j_1s)
{
    unsigned char i_1s;
    while(j_1s--)
        for(i_1s=0;i_1s<200;i_1s++);
}
void int_0() interrupt 0
{
    m_mode=~m_mode;
}

```

5.3.4 整机调试

进行功能调试的时候应该是一个模块一个模块地测试，首先是小车能自动循迹，然后是小车的壁障、探测金属，最后是无线遥控功能。

(1) 自动循迹调试

在使用循迹模块时，先确保单片机控制电路的正常使用，用万用表检查单片机控制电路的各线路是否有短路情况，确保单片机控制电路可以使用后，循迹模块也要先保证有信号输出，需要注意的是，由于电路多，驱动电路中的稳压 5V 中的稳压管输出的电流是比较小的，接入电源测试的时候电流输入是 2A 左右，因此有必要的就把 MC78M05 换成 LM7805，以免电流过大把单片机损坏了。调试的时候需要注意一个问题就是避免小车在昏暗的情况下使用，因为循迹模块中的光电传感器会对黑暗的情况无法准确判断黑白线，导致小车无法正常循迹。在烧写程序到单片机中前，把编写好的程序进行软件仿真，屏蔽掉其他功能的程序，调试好循迹功能后烧写到单片机中测试，测试成功之后再测试其他功能。为了小车在转弯的时候行驶快些，在向左转的时候，小车的左履带向后退，右履带向左行驶，行驶的时候小车是全速行驶的，当然也可以采用 PWM 脉宽调制的方式使小车调好速度走得更舒畅。

(2) 壁障和探测金属调试

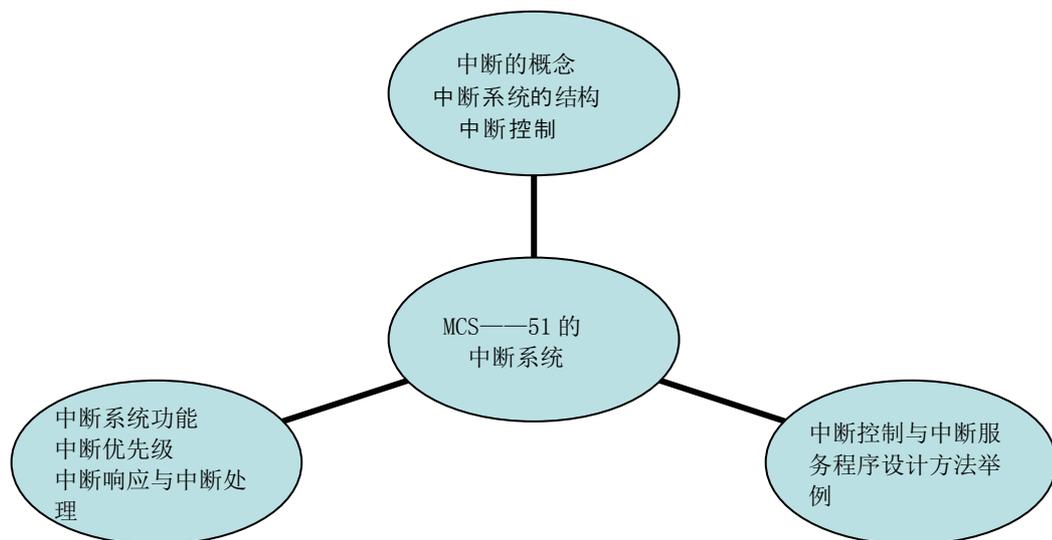
壁障功能是采用现有的模块调试的，所以也比较简单，输出的信号只是高低电平，如果有条件就可以自己采用分立元件制作一个。壁障功能也另外制作了一个报警器，不过这是报警器是不经过单片机控制的，需要一个驱动电路带动蜂鸣器响应，这个电路跟金属探测器的报警电路驱动一样，只是把壁障模块输出的信号直接接到报警驱动电路上，当小车遇到障碍是就有个高电平输入到单片机中，同时也带动报警驱动电路中的蜂鸣器响。金属探测器的报警时经过单片机的，所以先让金属探测器发出什么样的声音就根据编程来实现了，这里是每隔 0.5 秒就发一次声音。金属探测器有个电位器，这个电位器是调好金属报警的临界的，把金属靠近制作好的电感线圈中，调到有相应为止，当然，如果把报警信号直接接入单片机中，这样就很难调试，我们把信号接入一个发光二极管中调试，直到有信号输出就行了，也可以用一个发光二极管作为报警信号，这样防止金属报警器使用时不稳定，可以再调试临界电位器，直到达到最佳效果。

(3) 无线遥控功能调试

无线遥控是涉及到高频的信号，因为小车的电源要求比较高，在调试的时候电池用得很快，一般

接的是稳压电源进行调试，在接入发射模块和接收模块的电源上要做好相应的滤波，防止遥控控制时不稳定。遥控发射的编码芯片向接收的解码芯片是没有地址码的，所以同样频率的发射模块会影响小车的接收问题。

重点串联



基础训练

技能训练

【课时建议】 4 课时

实训 5.1: 无线电遥控车的设计

实训 5.2: 遥控车制作

思考与练习

一、填空题

- 1、外部中断 1 的中断入口地址为 _____， 定时器 1 的中断入口地址为_____。
- 2、若 (IP) =00010100B，则优先级最高者为_____， 最低者为_____。
- 3、8051 单片机响应中断后，产生长调用指令 LCALL，执行该指令的过程包括：首先把_____的内容压入堆栈，以进行断电保护，然后把长调用指令的 16 位地址送_____，使程序执行转向 _____中

的中断地址区。

二、判断题

- 1、8051 单片机对外中断请求源响应的最快响应时间为 3 个机器周期。()
- 2、各中断源发出的中断请求信号，都会标记在 8051 单片机的 TMOD 寄存器中。()

三、单选题

- 1、中断查询确认后，在下列各种 8051 单片机运行情况下，能立即响应的是()
A、当前正在进行高优先级中断处理
B、当前正在执行 RETI 指令
C、当前指令时 DIV 指令，且正处于取指令的机器周期
D、当前指令时 MOV A, R3
- 2、ORG 0003H LJMP2000H ORG 000BH LJMP3000H 当 CPU 响应外部中断 0 后, PC 的值是()
A. 0003H B. 2000H C. 000BH D. 3000H
- 3、若 MCS-51 中断源都编程为同级, 当他们同时申请中断时 CPU 首先响应()
A. 外部中断 0 B. 外部中断 1 C. T1 D. T0
- 4、控制串行口工作方式的寄存器是()
A. TCON B. PCON C. SCON D. TMOD
- 5、要使 MCS-51 能响应定时器 T1 中断，串行接口中断，它的中断允许寄存器 IE 的内容应是()
A. 98H B. 84H C. 42H D. 22H

四、计算题

- 1、已知一单片机系统的外接晶体振荡器的振荡频率为 11.059MHz，请计算该单片机系统的拍节 P、状态 S、机器周期所对应的时间是多少？指令周期中的单字节双周期指令的执行时间是多少？

五、论述题

1. 51 系列单片机具有几个中断源，分别是如何定义的？其中哪些中断源可以被定义为高优先级中断，如何定义？
- 2、各中断源对应的中断服务程序的入口地址是否能任意设定？
- 3、如果想将中断服务程序放置在程序存储区的任意区域，在程序中应该作何种设置？请举例加以说明。

课后习题答案

1、填空题

- 1、0013H 001BH
- 2、外部中断 1 定时器 T1
- 3、PC PC 程序存储器

2、判断题

- 1、对 2、错

三、单选题

- 1、D 2、B 3、A 4、C 5、A

四、计算题

答： $P = 1 / f = (1 / 11.059 \times 10^6) = 0.09 \times 10^{-6} \text{s} = 90 \text{ns}$ $S = 2P = 2 \times 90 \text{ns} = 180 \text{ns}$ 机器周期 = $12P = 12 \times 90 \text{ns} = 1.08 \mu\text{s}$ 双指令周期 = $2 \times$ 机器周期 = $2.16 \mu\text{s}$

五、论述题

- 1、答：具有 5 个中断源，分别是外部中断 INT0 和外部中断 INT1、定时器溢出中断 0 和定时器溢出中断 1 以及 串行中断。通过对中断优先级寄存器 IP 的设置，每个中断源都可以被定义为高优先级中断。
- 2、答：各中断源的入口地址已经在中断地址区中被定义了，不能任意设定。
- 3、答：如果要将中断服务程序放置在程序存储区的任意区域，在程序中要通过在中断地址区的对应地址上设置跳转指令才可实现对中断服务程序的执行。