

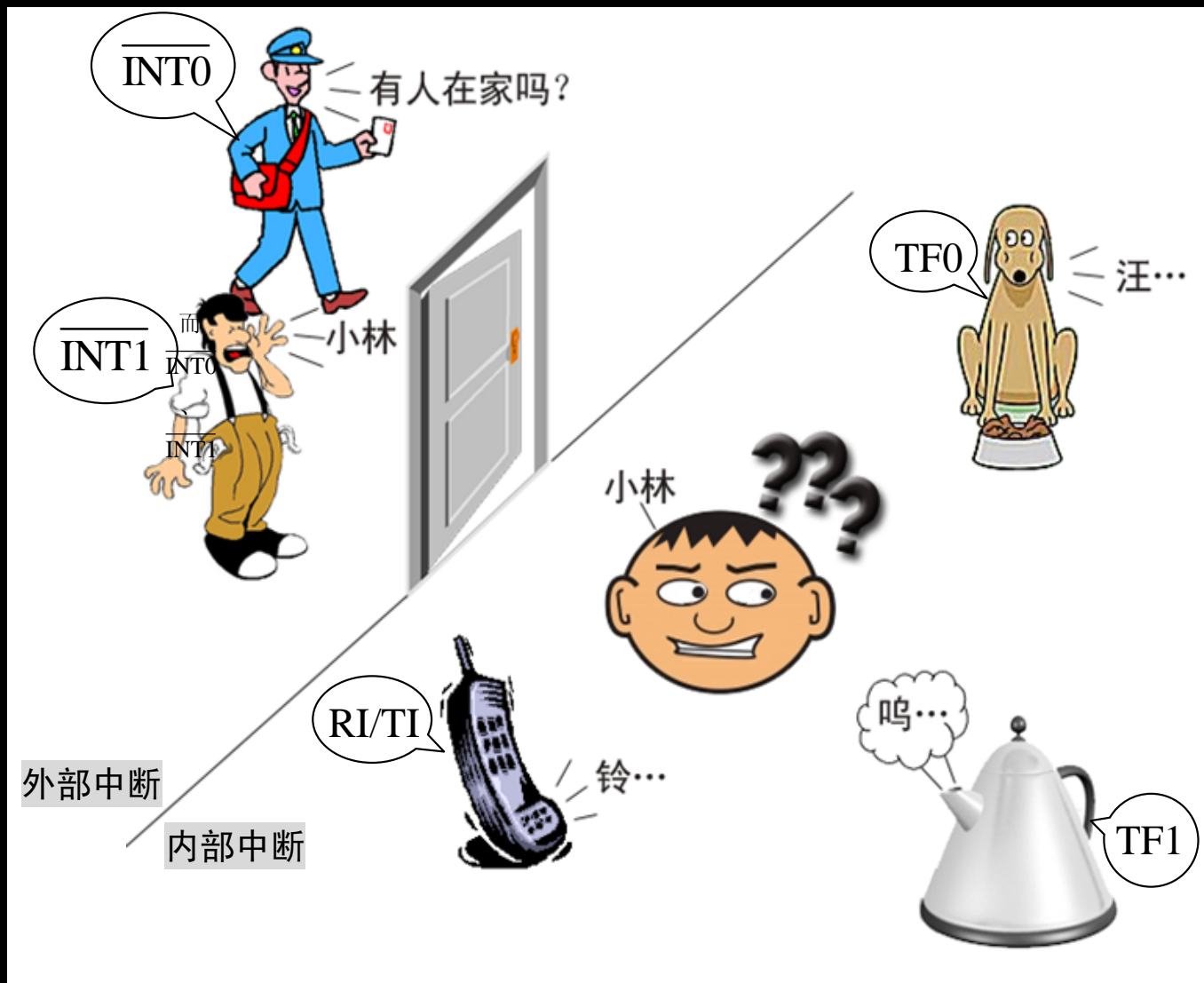
第11章 程序设计的魅力——中断控制

欢迎访问 电路飞翔网

<http://www.circuitfly.com> 获取更多信息

- 11.1 中断实例及中断的控制者
- 11.2 邮递员和朋友——外部中断
- 11.3 小狗和烧水壶——Timer 0和Timer 1中断
- 11.4 电话铃响——串行通信中断
- 11.5 中断的魅力和优先级控制
- 11.6 实例点拨——旋转的时钟

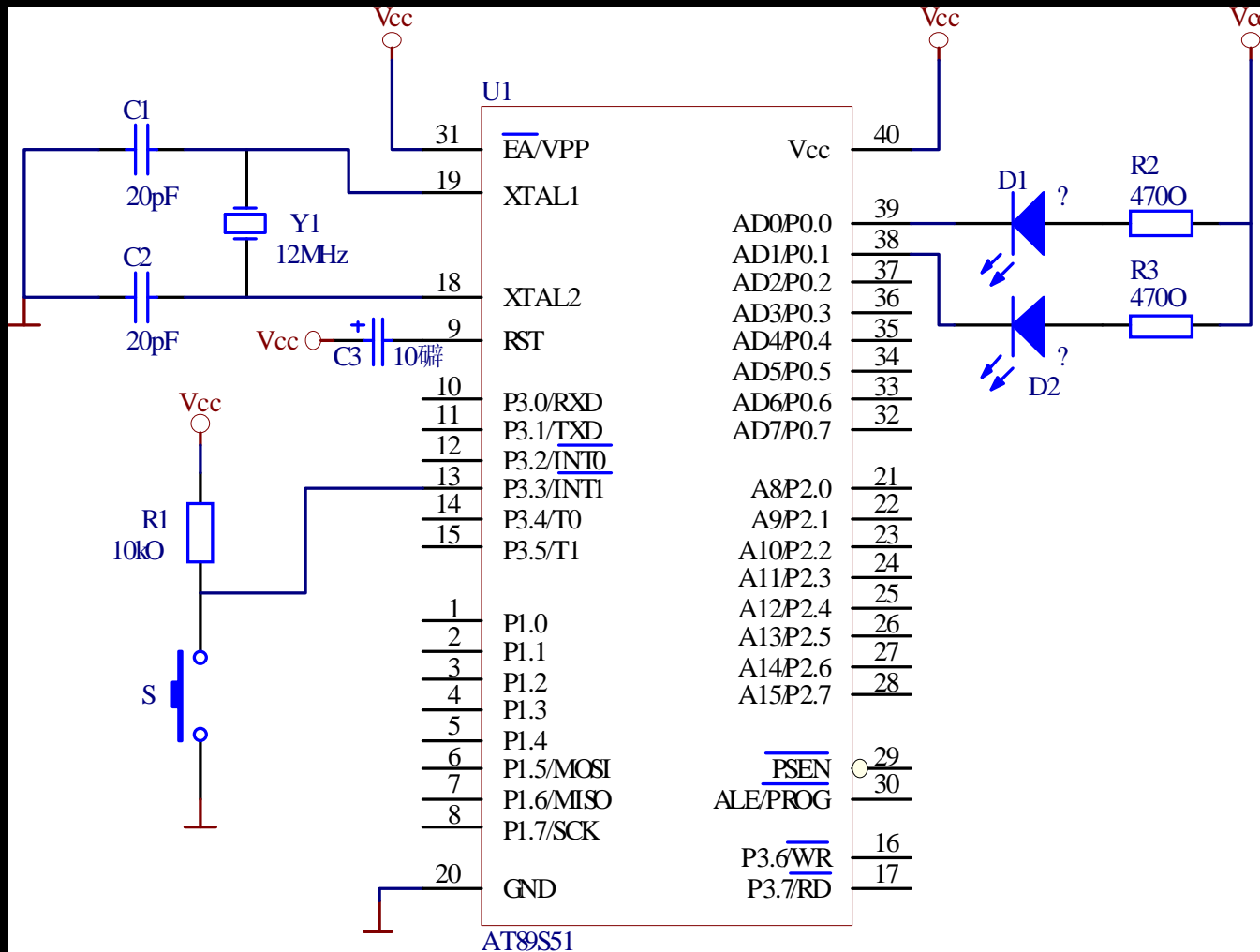
11.1 中断实例及中断的控制者



INT0、INT1、TF0、TF1、RI/TI这5个中断源在单片机中称为：
外部中断0、外部中断1、Timer 0中断、Timer 1中断、串行通信中断。

11.1 中断实例及中断的控制者

• 11.1.1 什么是中断



一个开关S和一个上拉电阻R1组成了一个外部中断信号发生器：当开关S闭合时，P3.3=0，外部中断产生。

11.1 中断实例及中断的控制者

- 11.1.1 什么是中断

例：开关中断产生后，出现绿灯闪一下，红灯闪一下。

ORG 00H ; 起始地址00H

; 下面是主程序段，点亮绿色发光二极管D1

MAIN:

MOV IE, #84H ; 使能外部中断1

GREEN:

CLR P0.0 ; 点亮绿色发光二极管D1

JMP GREEN ; 循环

11.1 中断实例及中断的控制者

• 11.1.1 什么是中断(实例软件)

； 中断服务子程序段， 熄灭绿色发光二极管D1， 点亮红色D2

ORG 13H ; 中断服务子程序起始地址为13H

EXT1_RED:

SETB P0.0 ; 熄灭绿色发光二极管D1

CLR P0.1 ; 点亮红色D2

D1: MOV R4, #200 ; 延时程序100ms

D2: MOV R5, #248

DJNZ R5, \$

DJNZ R4, D2

SETB P0.1 ; 熄灭红色发光二极管D2

RETI ; 中断服务子程序结束

END ; 程序结束

11.1 中断实例及中断的控制者

- 11.1.2 中断服务的指挥官——中断向量表

中断向量表（interrupt vector table），向量即“取向”的意思，例如外部中断0发生时，单片机会到程序存储器的0003H中寻找中断服务子程序来执行。

中 断 源	向 量 地 址	中断标志位
外部中断（ $\overline{\text{INT0}}$ 管脚）	0003H	IE0
Timer 0中断	000BH	TF0
外部中断（ $\overline{\text{INT1}}$ 管脚）	0013H	IE1
Timer 1中断	001BH	TF1
串行通信中断	0023H	TI/RI

11.1 中断实例及中断的控制者

• 11.1.3 中断响应与处理过程

- ① 立即暂停正在执行的任何程序，并把下一条将要执行的指令地址压入堆栈中。
- ② 保存所有中断状态（但不在堆栈中）。
- ③ 根据中断向量表找到中断对应的地址。
- ④ 单片机到该地址上开始执行中断服务子程序，直到遇到指令**RETI**为止。
- ⑤ 遇到**RETI**指令后，从堆栈中弹出在中断产生时保存的将要执行的下一条指令的地址到程序计数器**PC**中，单片机开始从**PC**指示的地址继续执行程序。

11.1 中断实例及中断的控制者

- 11.1.3 中断响应与处理过程

问题：

每一个中断服务子程序的存放空间都非常有限，例如，外部中断0的中断向量为0003H，而Timer 0中断向量地址为000BH，可见外部中断0的中断服务子程序只有0003H~000AH这8个字节的存储空间来存放，这8个字节的空间实在也放不了几条指令。

如果单片机同时使能了多个中断，并且每一个中断服务子程序都按中断向量表中的地址开始存放，就很有可能出现一个中断服务子程序占据其他中断服务子程序空间的问题。

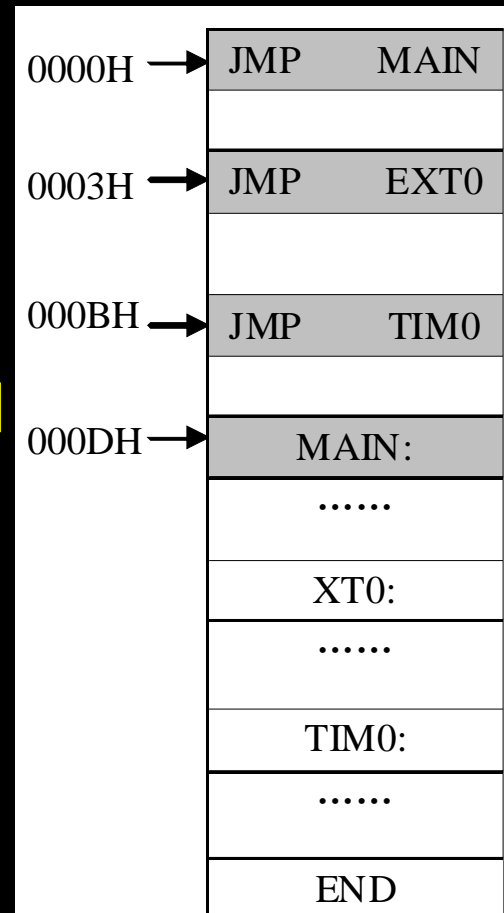
解决办法：JMP 指令

11.1 中断实例及中断的控制者

• 11.1.3 中断响应与处理过程

```

ORG 00H    ; 起始地址00H
JMP MAIN   ; 跳转到MAIN
ORG 03H    ; 外部中断0的起始地址03H
JMP EXT0   ; 跳转到EXT0
ORG 0BH    ; Timer 0中断的起始地址0BH
JMP TIM0   ; 跳转到TIM0
  
```



11.1 中断实例及中断的控制者

• 11.1.4 中断的控制者——中断使能寄存器IE

单片机上电复位之后，默认所有中断都被关闭。为了使单片机能对中断进行响应，需要对中断使能寄存器IE进行操作。IE的初始化过程即是使用MOV指令向其载入控制字。

7	6	5	4	3	2	1	0
EA	保留	保留	ES	ET1	EX1	ET0	EX0

EA 中断使能。

保留 不要访问该位。

ES 串行口中断使能。

ET1 Timer 1中断使能。

EX1 外部中断1使能。

ET0 Timer 0中断使能。

EX0 外部中断0使能。

11.2 邮递员和朋友——外部中断

• 11.2.1 外部中断 $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$

(AT89S51单片机的两个外部中断触发端口)

外部中断 0 触发端口	10	P3.0/RXD	AD5/P0.5	33
	11		AD6/P0.6	32
外部中断 1 触发端口	12	P3.1/TXD	AD7/P0.7	
	13	P3.2/ <u>INT0</u>		21
	14	P3.3/INT1	A8/P2.0	22
	15	P3.4/T0	A9/P2.1	23
		P3.5/T1	A10/P2.2	24
			A11/P2.3	

这两个外部中断对应的中断向量的地址分别为0003H和0013H，分别由IE中的EX0和EX1位来使能/屏蔽。

11.2 邮递员和朋友——外部中断

- 11.2.2 外部中断的电平触发（ level-triggered or level-activated interrupt ）
 - ✓ 在使能外部中断后，如果在 $\overline{INT0}$ 或 $\overline{INT1}$ 口上出现低电平就会触发外部中断0或外部中断1，单片机会立即停止正在执行的程序，执行中断服务子程序，是单片机复位之后外部中断的默认触发方式。
 - ✓ 在单片机进入执行中断服务子程序时，在执行到中断服务子程序结束指令**RETI**之前，这个低电平必须撤走，否则将会使单片机执行完**RETI**指令后再次进入中断服务子程序。

11.2 邮递员和朋友——外部中断

- 11.2.2 外部中断的电平触发（ level-triggered or level-activated interrupt ）
 - ✓ 一般为了保证外部中断的产生，低电平的时长至少需要4个机器周期，若晶振式12MHz,则：

1 个机器周期

1 μ s

4 个机器周期

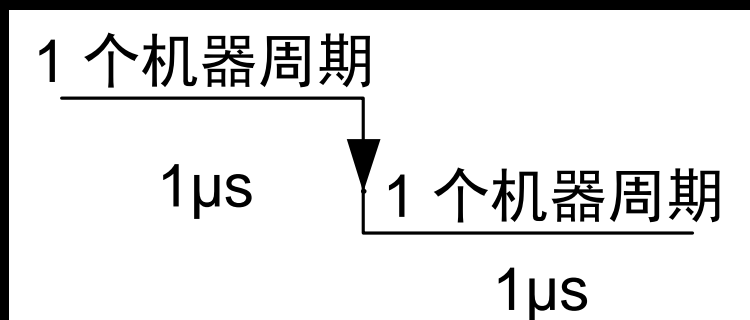
4 \times 1 μ s

至 $\overline{\text{INT0}}$ 或 $\overline{\text{INT1}}$ 端口

低电平触发外部中断的保持时间

11.2 邮递员和朋友——外部中断

- 11.2.3 外部中断的边沿触发（edge-triggered or edge-activated interrupt）
 - ✓ 至少1个机器周期的高电平，且向低电平跳变之后至少保持1个机器周期的低电平，如果晶振频率为12MHz， 则：



11.2 邮递员和朋友——外部中断

- 11.2.3 外部中断的边沿触发（edge-triggered or edge-activated interrupt）

边沿触发方式需要指令来设置寄存器TCON。

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

与定时/计数器有关

IE1外部中断1边沿标志位

IT1外部中断1方式控制位

（IT1清0电平触发/置1边沿）

IE0外部中断0边沿标志位

IT0外部中断0方式控制位

（IT1清0电平触发/置1边沿）

11.2 邮递员和朋友——外部中断

- 11.2.3 外部中断的边沿触发（edge-triggered or edge-activated interrupt）

① 电平触发方式下，若按下开关S不放，端口将不断获得低电平，中断不断产生，中断服务子程序被重复执行；边沿触发方式，只对端口的电平跳变有反应，若也不放开开关S，并不会连续地产生中断。

② 电平触发方式下，若单片机正在执行中断服务子程序时，外部中断再次发生，电平触发信号不会被锁存，所以单片机不会响应第2次中断。而在边沿触发方式下有所不同，单片机正在执行中断服务子程序，如果中断发生，电平跳变产生的中断触发信号保存在TCON寄存器中的IE0或IE1中，单片机再次检测IE0或IE1位，又进入中断服务程序中。

注意：在单片机执行中断服务子程序时，电平触发方式会“漏掉”新的中断，而边沿触发方式会“保留”新的中断。

11.3 小狗和烧水壶

——Timer 0和Timer 1中断

- 问题：前面通过检测Timer溢出标志位TFx来判断计时是否完成。在判断的过程中，单片机除了循环判断之外一般不做任何操作。如果在计时的同时又需要执行别的程序该怎么办呢？
 - 解决办法：当通过中断使能寄存器IE使能Timer中断，设置好Timer的工作参数，启动Timer之后单片机就可以去做别的操作。Timer溢出时，单片机将会自动停下正在执行的任何程序而进入Timer中断服务子程序中执行。
- 两个中断的中断向量地址分别为000BH和001BH。

11.3 小狗和烧水壶

——Timer 0和Timer 1 中断

- 例：利用Timer 0中断产生50Hz矩形波

ORG 00H ; 起始地址00H

JMP MAIN ; 跳转到MAIN

ORG 0BH ; Timer 0中断服务子程序起始地址为0BH

JMP TIM_LOW ; 跳转到TIM_LOW

; 下面是主程序段，初始化

MAIN:

MOV IE, #82H ; 使能Timer 0中断

MOV TMOD, #01H ; 设置Timer 0工作在模式1下

TIMER0_LOAD:

MOV TH0, #0B1H ; 往TH0中载入计数初始值

MOV TL0, #0E0H ; 往TL0中载入计数初始值

SETB TR0 ; 启动Timer 0

SET_HIGH:

SETB P1.0 ; P1.0=1

JMP SET_HIGH ; 循环

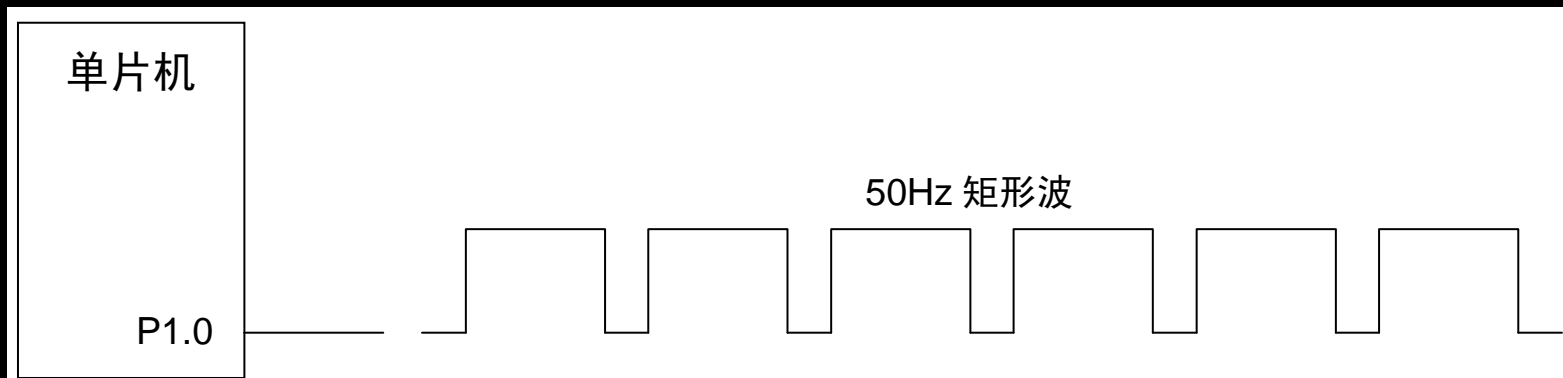
11.3 小狗和烧水壶

——Timer 0和Timer 1 中断

; 下面是Timer中断服务子程序

TIM_LOW:

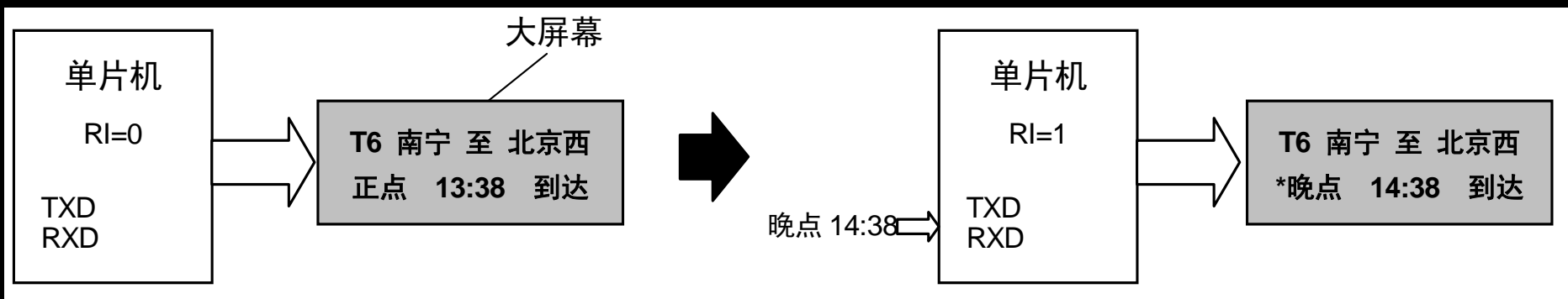
CLR	P1.0	; P1.0=0
MOV	TH0, #0B1H	; 往TH0中载入计数初始值
MOV	TL0, #0E0H	; 往TL0中载入计数初始值
RETI		
END		



11.4 电话铃响——串行通信中断

- ✓ 第十章中串行口发送和接收过程存在一个问题，就是单片机在判断**TI**和**RI**位时只能“一心一意”地进行而无法再进行其他操作。
- ✓ 串行通信中断的出现较好地解决了系统中既需要单片机响应串行通信的操作，也同时兼顾其他操作。
- ✓ 串行通信中断的中断向量地址为**0023H**。

11.4 电话铃响——串行通信中断

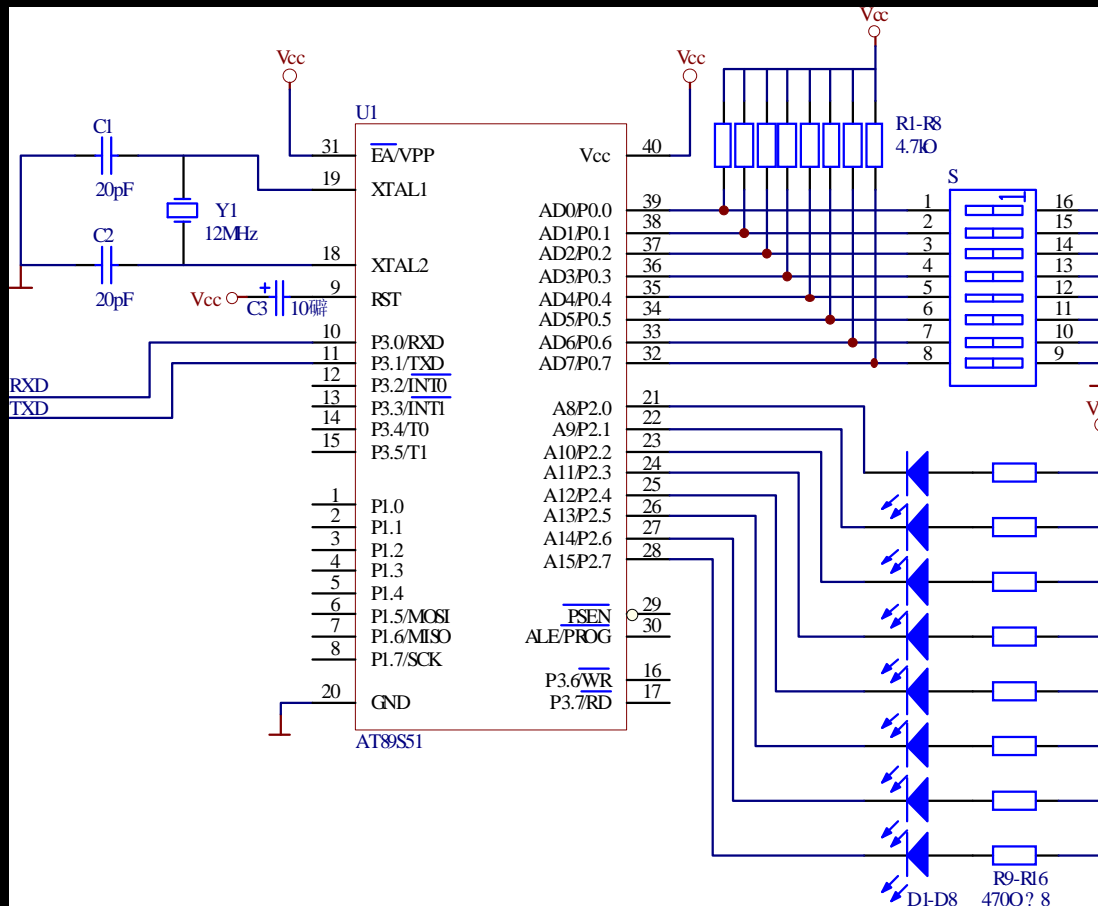


单片机串行通信中断（例）

- ✓ 单片机被设置工作在串行通信中断时，可以照常执行主程序。当接收或发送数据完成时，**RI**或**TI**会被硬件置**1**而产生串行通信中断，单片机则响应中断。
- ✓ 因为**RI**或**TI**的置**1**都会使单片机中断，程序要知道是发送数据完成的**TI=1**产生的中断或是串行口接收到数据的**RI=1**产生的就需要在中断服务子程序中先对**TI**和**RI**进行判断。

11.4 电话铃响——串行通信中断

- 例：将P0口上DIP开关产生的8位状态信号从串行口TXD发送出去，并由P2口上发光二极管显示。



串行通信中断的演示电路图

11.4 电话铃响——串行通信中断

- 例：串行通信中断的演示程序

```
ORG    00H    ; 起始地址00H
JMP                      MAIN ; 跳转到MAIN
ORG    23H    ; 串行中断服务子程序起始地址为23H
JMP                      SERIAL ; 跳转到TIM_LOW
; 下面是主程序段，初始化并发送数据
```

11.4 电话铃响——串行通信中断

- 例：串行通信中断的演示程序

MAIN:

```
MOV    IE, #90H           ; 使能串行中断
MOV    SCON, #50H         ; 设置串行口工作在模式1下
MOV    TMOD, #20H         ; 使用Timer 1工作在模式2下
MOV    TH1, #0F3H; 波特率2400（晶振频率=12MHz）
MOV    TL1, #0F3H
SETB   TR1                ; 启动Timer 1
```

SEND_DATA:

```
MOV    A, P0              ; 将P0口状态数据载入ACC
MOV    SBUF, A            ; 载入缓冲区以发送
MOV    P2, A              ; 在P2口上的发光二极管显示
JMP    SEND_DATA          ; 循环检测P2口状态并发送
```


11.4 电话铃响——串行通信中断

- 例：串行通信中断的演示程序

SERIAL: ; 这里是串行中断服务子程序

JB TI, SENT ; 如果TI=1, 说明是发送完成产生的中断,
; 跳到 **SENT**段将TI标志清0后返回主程序

MOV A, SBUF ; 如果TI=0, 说明是接收到数据产生的中断,
; 于是将缓冲区中的数据载入ACC中

MOV P2, A ; 在P2口上的发光二极管显示

CLR RI ; 清标志位RI

RETI ; 返回主程序

SENT:

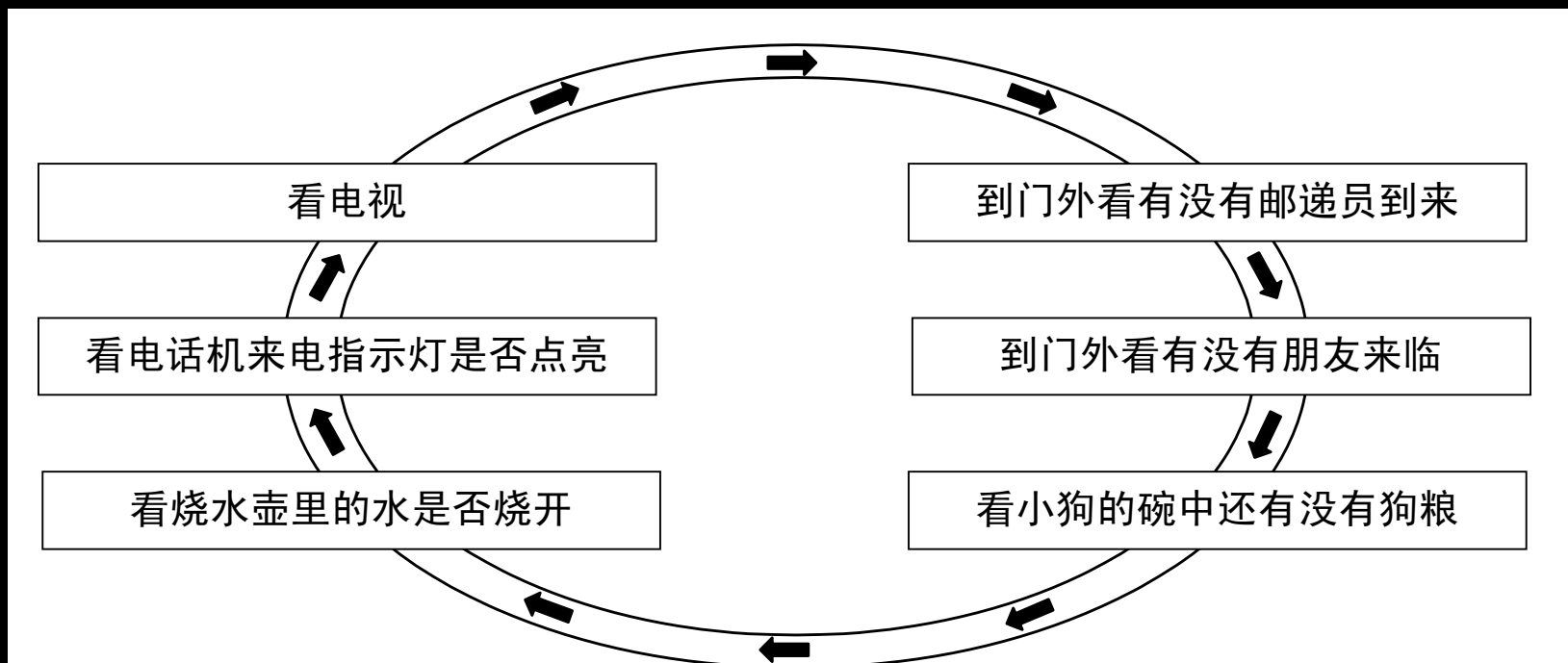
CLR TI ; 发送完成, 清标志位TI

RETI ; 返回主程序

END ; 程序结束

11.5 中断的魅力和优先级控制

• 11.5.1 中断的魅力——轮询与中断的比较



小林的循环关注

单片机的程序资源都耗费在访问设备的状态上，如果设备响应率比较低，单片机的程序执行效率将很低。

11.5 中断的魅力和优先级控制

• 11.5.1 中断的魅力——轮询与中断的比较

- ✓ 在中断下，单片机同样能服务多个设备，等待某个设备产生中断时再去处理。在设备没有产生中断时单片机还可以专心地执行其他程序。
- ✓ 中断的好处：
 - ① 单片机可以选择特定的中断进行响应和处理。可通过中断使能寄存器IE相应的位来使能系统希望响应和处理的 interrupt，而屏蔽那些不需要响应和处理的 interrupt。轮询模式下，单片机必须响应和处理每一个设备。
 - ② 如果有多个中断同时发生，单片机可以根据程序的设定来优先响应和处理某一 interrupt。而轮询模式下，单片机逐一对设备进行访问，没有优先顺序之分。

11.5 中断的魅力和优先级控制

• 11.5.2 中断的优先级控制

单片机能根据设置（中断优先级控制寄存器**IP**）来优先响应和处理某一中断，**IP**寄存器位于特殊功能寄存器区的**B8H**上。

7	6	5	4	3	2	1	0
保留	保留	保留	PS	PT1	PX1	PT0	PX0

置1则优先

保留 不要访问保留位，也不要往保留位写数据。

PS 串行口中断优先。 **PT1** **Timer 1**中断优先。

PX1 外部中断1优先。 **PT0** **Timer 0**中断优先。

PX0 外部中断0优先。

11.5 中断的魅力和优先级控制

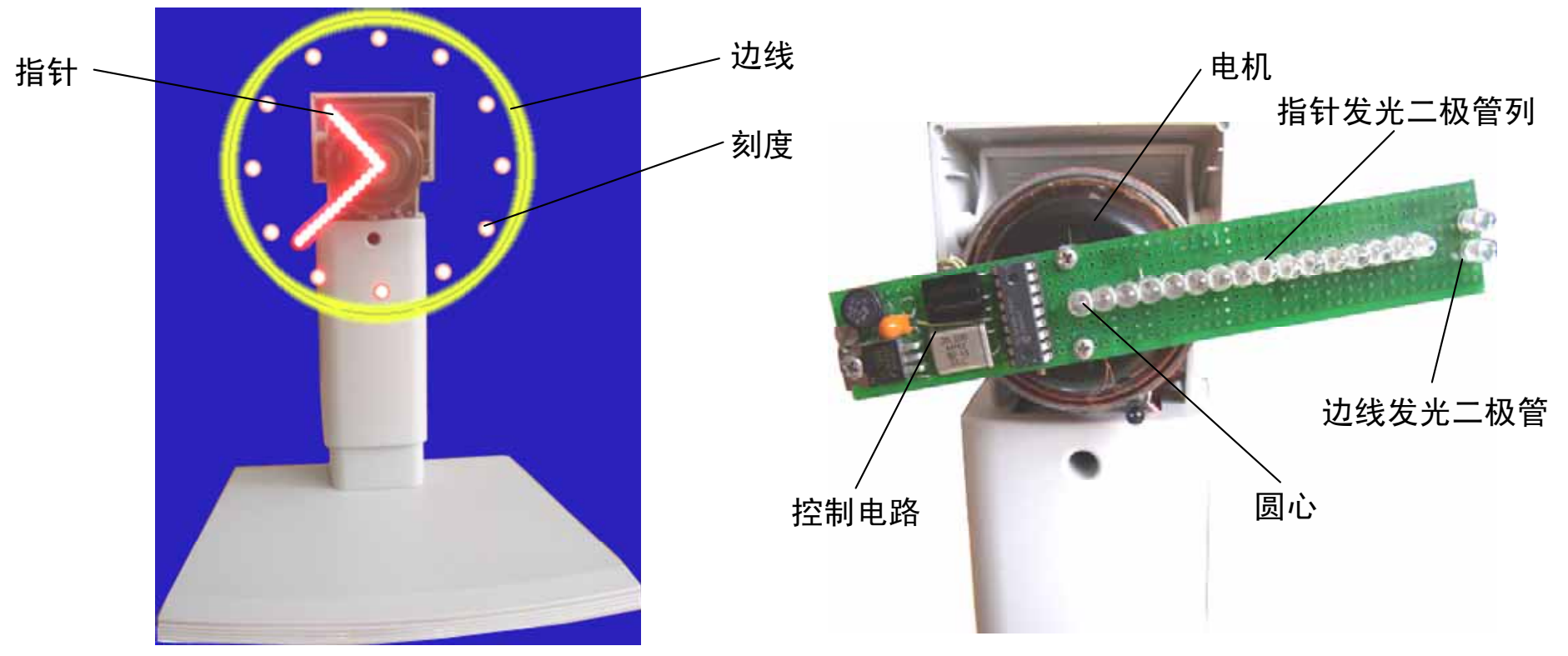
• 11.5.2 中断的优先级控制（默认的中断优先级）

优 先 级	中 断 源
最高  最低	外部中断 0 $\overline{\text{INT0}}$
	Timer 0 中断 TF0
	外部中断 1 $\overline{\text{INT1}}$
	Timer 1 中断 TF1
	串行通信中断 RI/TI

✓如果想调整表所示的默认中断优先级，可以将中断优先级控制寄存器**IP**中相应的位置**1**即可。

✓如果程序中把两个或两个以上的中断优先级位都置**1**，单片机又会根据表中描述的中断优先级来服务。

11.6 实例点拨——旋转的时钟



(a) 旋转的时钟

(b) 时钟只有一列发光二极管

11.6 实例点拨——旋转的时钟

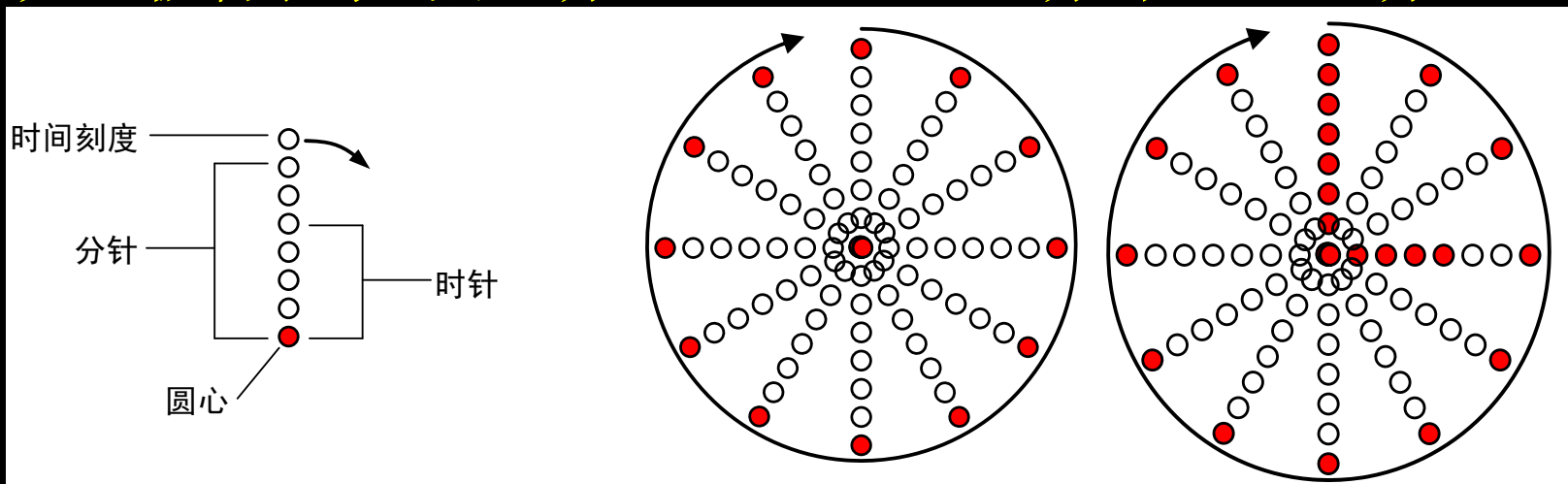
• 11.6.1 原理分析

- ✓ 人眼具有视觉暂留（**visual persistence**）的错觉，即无法区分间隔小于0.1秒的图像。
- ✓ 有8支发光二极管排成一行，最内侧的发光二极管为圆心，它们绕着圆心旋转，最外侧的发光二极管显示时间刻度，当时针在12点时我们假设角度为 0° ，则每个小时时针之间的角度为 $30(360/12)$ 。于是当这一列发光二极管每旋转 30° ，最外侧的发光二极管就点亮一个瞬间以呈现出时间刻度。这样，如果在0.1秒之内这列发光二极管能旋转完一圈，则人眼就会产生错觉，而把先后产生的时间刻度连成12个完整的时间刻度。

11.6 实例点拨——旋转的时钟

• 11.6.1 原理分析

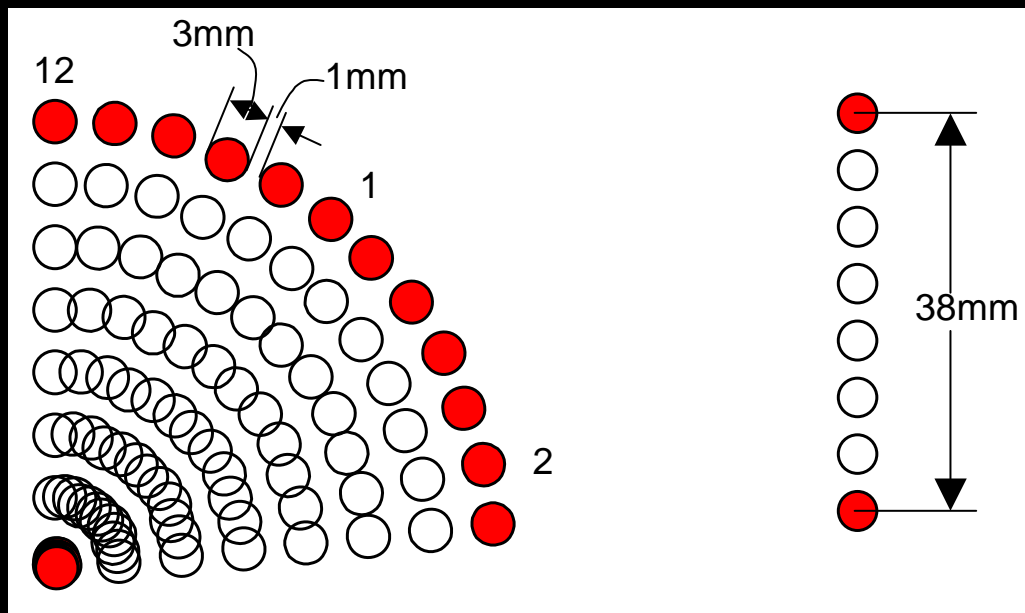
- ✓ 分针较时针长，假设分针用7支发光二极管来显示，时针用5支发光二极管来显示。要显示3点整，在时间刻度的显示基础上，控制在 0° 时点亮7支发光二极管以显示分针指在12点位置上；然后当发光二极管转过 90° 时再显示5支以显示时针指在3点位置上，由于发光二极管在不断地旋转，在0.1秒内重复点亮12点位置上的7支和3点位置上的5支发光二极管，直到1分钟以后在 6° 的位置上点亮7支发光二极管表示3点01分（ $360^\circ \div 60 \text{ (分钟)} = 6^\circ / \text{分}$ ）



11.6 实例点拨——旋转的时钟

• 11.6.1 原理分析（具体的设计过程）

- ✓ 计算旋转时钟的周长和半径。选用直径为3mm的发光二极管组成一列，这样最外侧的发光二极管旋转一周经过的显示位置应该有60个。为了在显示时每分钟之间有一个间隔，我们在每分钟之间引入一个1mm空隙，于是得到旋转的周长 $C=60 \times 3\text{mm} + 59 \times 1\text{mm} = 239\text{mm}$ ，可得半径 $R = C / 2\pi = 38\text{mm}$ ，将8支发光二极管按38mm平均间距排成一列。



11.6 实例点拨——旋转的时钟

- 11.6.1 原理分析（具体的设计过程）
 - ✓ 进行时间的计算。根据视觉暂留的原理，该列发光二极管在同一位置出现的时间间隔应该不大于0.1秒，即旋转周期 $T \leq 100\text{ms}$ 。为了计算方便，可取旋转周期 $T = 60\text{ms}$ 。这样每分钟刻度上发光二极管的点亮和熄灭的时间 $t = 60\text{ms} / 60 = 1\text{ms}$ 。在这1ms中，可设计发光二极管点亮 0.7ms （ $700\mu\text{s}$ ），熄灭 0.3ms （ $300\mu\text{s}$ ）

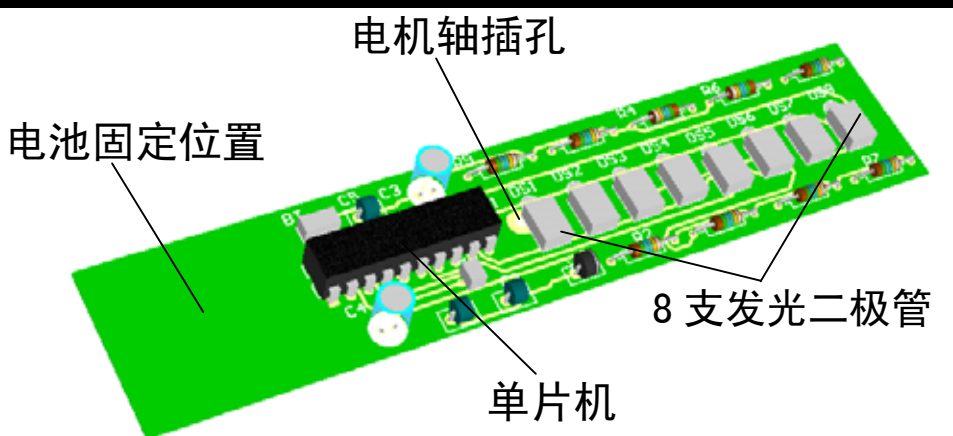
11.6 实例点拨——旋转的时钟

- 11.6.2 明确系统功能

电机带动一系列发光二极管绕轴旋转，单片机控制发光二极管在旋转过程中的相应位置上点亮以指示时间和刻度。与此同时，单片机内部进行时钟的计时操作，以控制旋转时钟显示正确的走时。

11.6 实例点拨——旋转的时钟

• 11.6.3 硬件设计

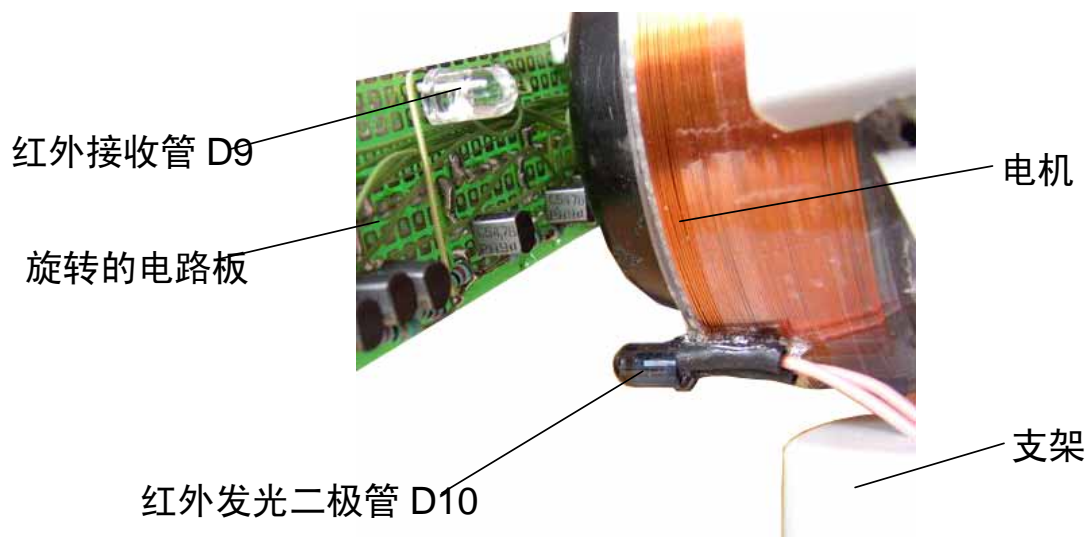


▷ 直流电机



△ 旋转时钟的电路板

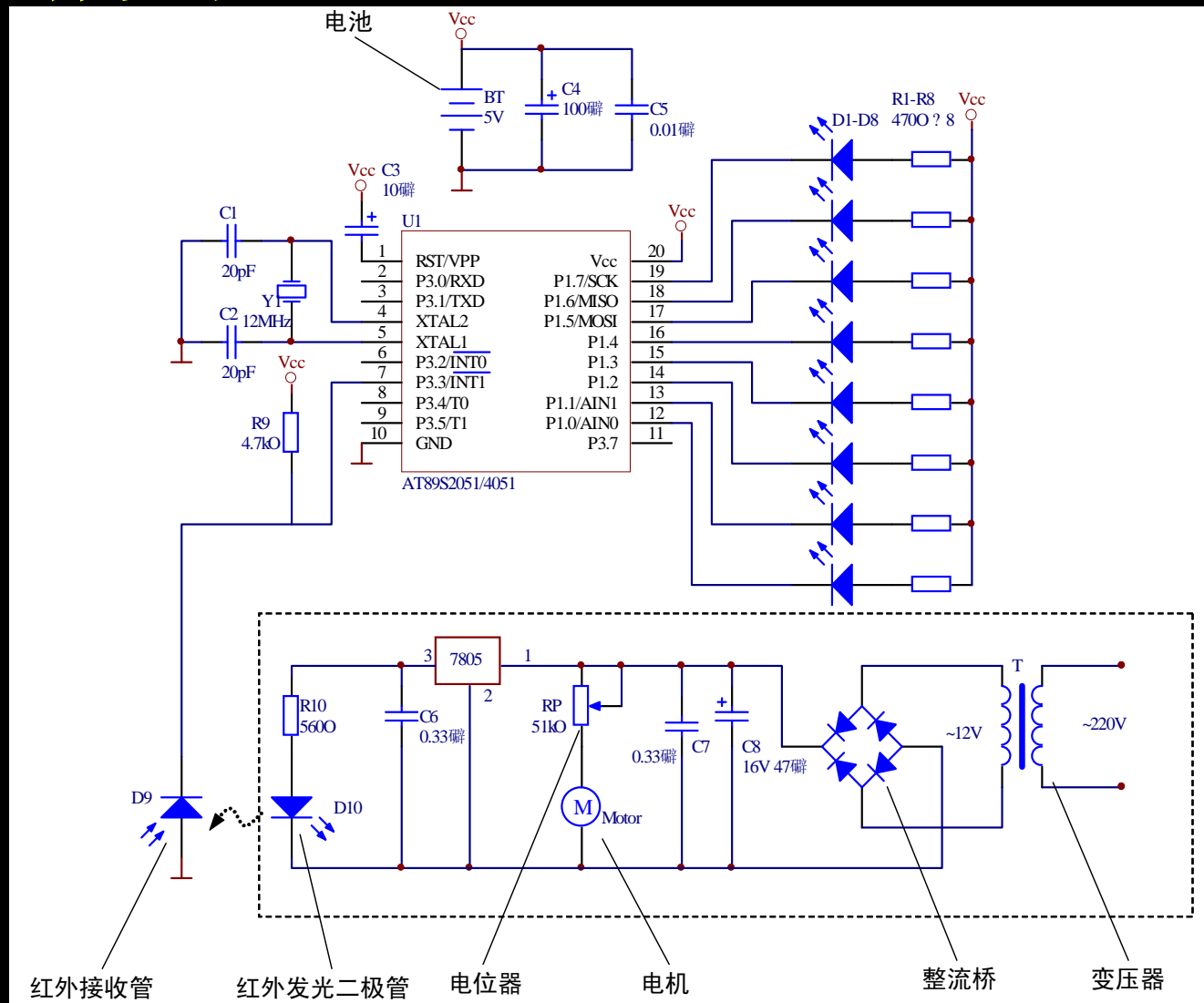
▷ 红外对管用于同步



11.6 实例点拨——旋转的时钟

• 11.6.3 硬件设计

旋转时钟系统电路图



11.6 实例点拨——旋转的时钟

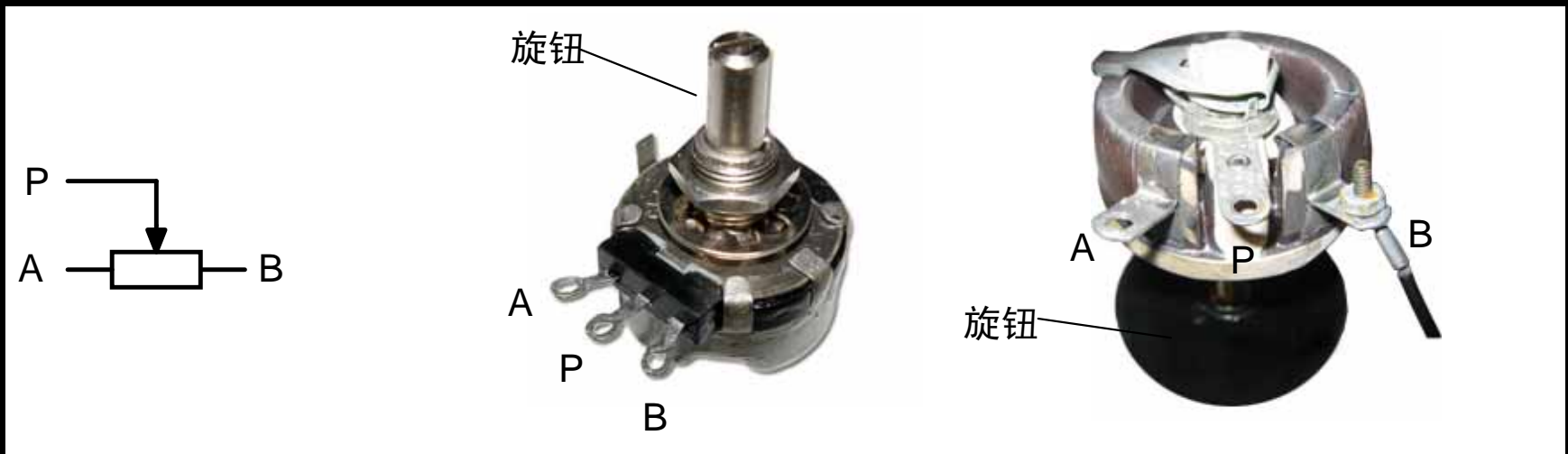
• 11.6.4 软件设计

- ✓ 使能单片机的两个中断——Timer 0中断和外部中断1，并设Timer 0的优先级较高。
- ✓ 启动Timer 0后，每50ms产生一次中断。
- ✓ 在Timer 0中断服务子程序TIM0中，R0计数器判断每50ms产生一次的中断是否到达20次，如果是计时1秒。R1是秒的计数器，如果R1计到60表明到达1分钟，则30H中的分钟值加1。
- ✓ 在计时的同时，外部中断1使能。每当红外对管对齐时，即每旋转一圈时中断服务子程序EXT1被执行。EXT1中将旋转一圈分成60份。根据60个不同的位置来判断是否显示时间刻度、分针和时针。
- ✓ (程序见书本)

器件介绍

1. 电位器:

A、B两端的阻值固定，等于电位器的标称值。中间引脚P与两端引脚之间的电阻值随着电位器的旋钮（**knob**）调节而改变。电位器外壳一般都会标注最大阻值和功率。大功率场合一般都使用绕线电位器。

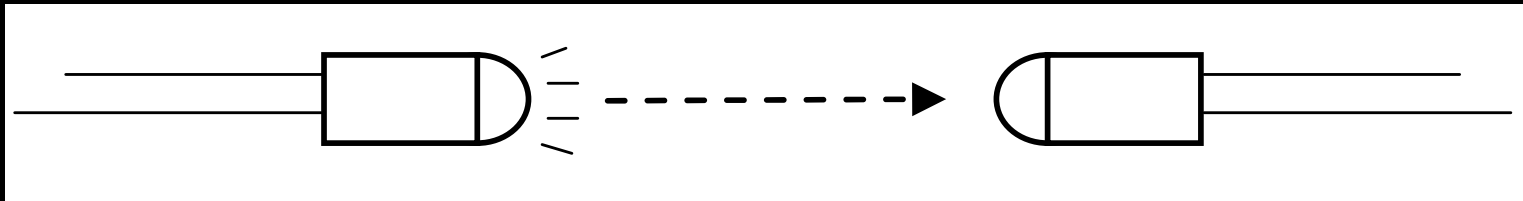


(a) 电路符号 (b) 一般电位器外观 (c) 绕线式电位器

器件介绍

2. 红外对管

红外对管（**IR transmitter and receiver**）是一种常用的光电器件。所谓对管就是由一支红外线发射管和红外线接收管组成。当发射管与接收管的发射与接收窗靠近对齐时，则接收管导通，一般来说，红外对管的有效距离为数米，如果想扩大感应距离可加装透镜。



红外对管的使用