**第6章 单片机的定时/计数器**

**本章重点：**

☆定时/计数器的4种工作方式的设置。

☆不同工作方式初值的计算方法。

**学习目标：**

☆定时/计数器的基本结构。

☆TMOD,TCON的设置。

☆初值的计算和基本程序编写方法。

在生活中处处可见定时/计数器的身影，小到电子手表，洗衣机的定时系统，大到工业设备的定时计数控制模块。这些往往需要一系列的定时，延时或对外部信号的计数等操作，而这些功能的实现通常需要以下几种方法实现。

方法1：软件法。通过循环函数来实现时间上的延时。无法做到精准的定时操作。这种方法优点是修改灵活编写容易，但是因为程序需要CPU计算会一定程度上占用CPU资源，影响总体的运行速度。由于不同CPU性能差异同一代码在不同机器运行差异大，容易引起误差不具有很好的移植性。

方法2：硬件法。完全通过硬件逻辑电路实现，优点运行速度快不占用CPU资源，定时/计数精确。但因为硬件固定修改需要对硬件内部的原件参数进行修改，因此不够灵活不具有很好的移植性。定时/计数时因硬件因素也有时长的限制。

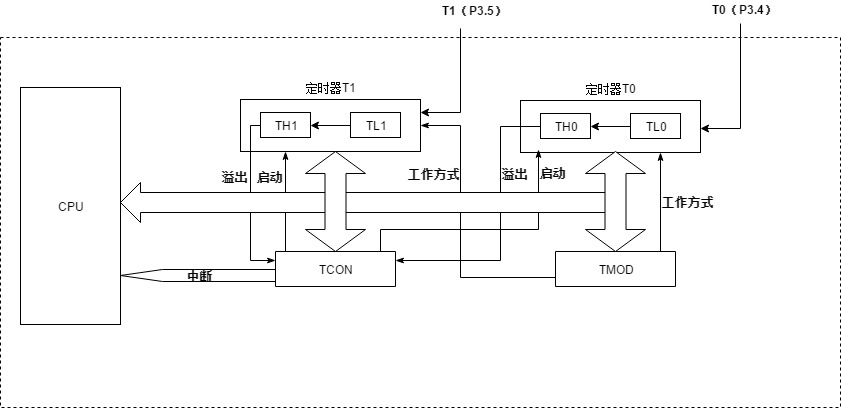
方法3：可编程的定时/计数器。这是以上两种方法的结合，利用软件的编程来实现对硬件控制的目的，通过中断和查询两种方式来实现定时和计数功能。这种方法使得定时和计数更为精确，并且可以方便的修改使用更加灵活。

本章节将详细介绍可编程的定时/计数器的原理，以及控制的方法。

**6.1 定时/计数器的结构和工作原理**

**6.1.1 定时/计数器的结构**

如图6-1为51单片机的定时/计数器的结构图。

图6- 1

由图我们可以看到有两个定时器T1和T0，而它们由特殊功能寄存器TCON来控制启动，由特殊功能寄存器TMOD来控制它们的工作方式，另外T1(P3.5)和T0(P3.4)分别为定时器T1和定时器T0的外部引脚可以用来接受外部的脉冲信号来计数。定时器T1和T0实质上是两个16位加1计数器，分别由高8位和低8位组成。通过一系列的设置自动加1，当计数器加满后则产生溢出控制特殊功能寄存器TCON的相应溢出标志位的改变，代表定时器工作结束并向CPU申请中断，引起CPU的响应。

而这个过程不需要CPU来控制完全由硬件实现因此计时精确节省了CPU的资源，并且我们可以通过修改程序定时器T1或T0的初始值来改变定时的时长从而实现灵活准确的控制定时和计数功能。

**6.1.2 定时/计数器的工作原理**

单片机内部一般自带一个振荡器，它经过片内12分频,产生单片机所必须的时钟频率，我们管这个振荡器的固有频率称为晶振（默认晶振 =12MHz）经过片内12分频而得到的时钟频率可以用来求出机器周期（12MHz晶振所产生的机器周期为1μs）具体计算将在后面提到。

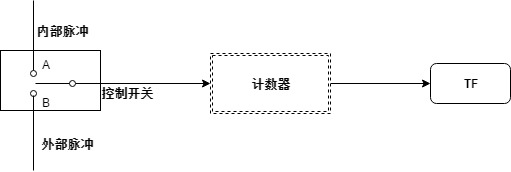


图6- 2

如图6-2,我们可以看到通过控制开关可以选择两种脉冲信号，即内部脉冲和外部脉冲。脉冲信号本质为电平的变化，当电平变化时就会使计数器加1从而达到对计数器充值的效果，当计数器数值超过容量即会溢出这时候计数器自动清0中断标志位变化（TF位）从而触发中断，引起CPU响应计时结束。

因此，我们如果需要一个特点的定时时长，就可以在开始计数之前通过软件在计数器内部初始化一个特定的初值，这样计数时就从当前的数值开始计数这样就可以达到控制任意的定时的时长。

**思考1：**这个时长真的是任意的么?任意时长是怎么实现的?

既然我们了解了定时的基本原理，那么我们就可以开始计数啦。如图6-3

，这个外部脉冲由图6-1中的T1(P3.5)和T0(P3.4)来获取，我们在每个机器周期对外部脉冲进行采样，当外部脉冲电平发生变化时，就控制计数器加1即通过对外部脉冲检测相邻机器周期的采样值是否相同来计数。

其实，定时和计数的最大区别在于获取脉冲的来源，定时器来自系统内部而计数器是对外部引进的脉冲来计数的。

**思考2：**计数器对外部脉冲有要求吗? 为什么?

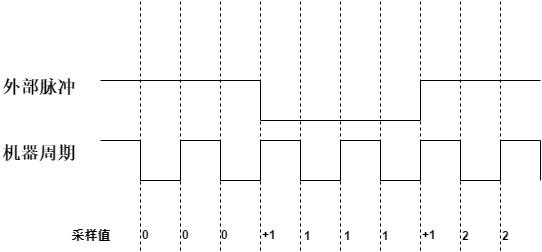


图6- 3

**解答1**：需要注意的是因为定时时长=[硬件容量()-初值(a)]机器周期()，所以定时时长要小于等于(n为定时器位数，注意不同的工作方式位数不同，具体区别将在后面提到)。

定时时长：

机器周期：

**解答2**：因为我们是在固定机器周期进行采样，所以分辨率最小为机器周期，若外部脉冲小于机器周期，我们将得不到准确的计数值。因此测量的外部脉冲需大于机器周期。

**6.2 定时/计数器的控制寄存器**

在第5章通过对中断对学习我们学会了通过一些特殊功能寄存器的设置来控制中断系统的工作，那么想必这章内容对于你将不会太难，对于定时器/计数器的设置也是通过对特殊功能寄存器TMOD和TCON的设置来控制它们的启停和中断申请的。

**6.2.1 TMOD寄存器**

**TMOD**工作方式寄存器（Timer/Conter Made Control）容量为一个字节，字节地址89H不可按位寻址。TMOD主要用于设置T0，T1的工作方式即选择不同功能的定时器/计数器，高4位控制T1的工作方式，低4位控制T0的工作方式。设置格式如表6-1。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **定时器T1** | | | | **定时器T0** | | | | |
| **GATE** | **C/T** | **M1** | **M0** | **GATA** | | **C/T** | **M1** | **M0** |
| **(89H)位7** | **位6** | **位5** | **位4** | **位3** | | **位2** | **位 1** | **位0** |
| 1.GATE：门控位（以定时器T0为例）  GATE =0 ：仅可通过TR0来控制定时器T0.  GATE =1 ：由外部引脚INT0和TR0来共同控制定时器T0。  2.C/T:功能选择位  C/T=0：定时器方式，以内部脉冲为信号。  C/T=1：计数器方式，以P3.4外部引脚输入脉冲为信号。  3．M1 M0：工作方式设置位（如下） | | | | | | | | |
| **M1 M0设置的4种工作方式** | | | | | | | | |
| **M1** | | **M0** | **工作方式** | | **功能** | | | |
| 0  0  1  1 | | 0  1  0  1 | 0  1  2  3 | | 13位定时/计数器  16位定时/计数器  8位自动重装定时/计数器  3种定时/计数器 | | | |

表6- 1

由上表我们可以看出定时器T0与定时器T1大同小异主要由三部分组成GATE，C/T ,M1 M0 ,这里以定时器T0为例，结合图6-4，GATE属于控制模块的一部分与TCON寄存器中的TR0和外部引脚INT0共同控制开关S1来达到控制定时器T0的启停。而C/T一方面连接了内部晶振，另一方面连接来外部引脚T0（P3.4）不难看出C/T具有控制信号来源的作用，通过对信号来源的选择我们可以很容易的切换定时器和计数器，M1 M2则是对不同功能定时/计数器进行选择对控制位，具体功能和设置见上表6-1.

需要注意的是，这4种工作方式中方式0，方式1和方式2对于定时器T0和定时器T1都适用，而方式3由于硬件的设计占用了T1的硬件资源因此只允许定时器T0使用，T1不能选用方式3若强行设置T1将停止工作。

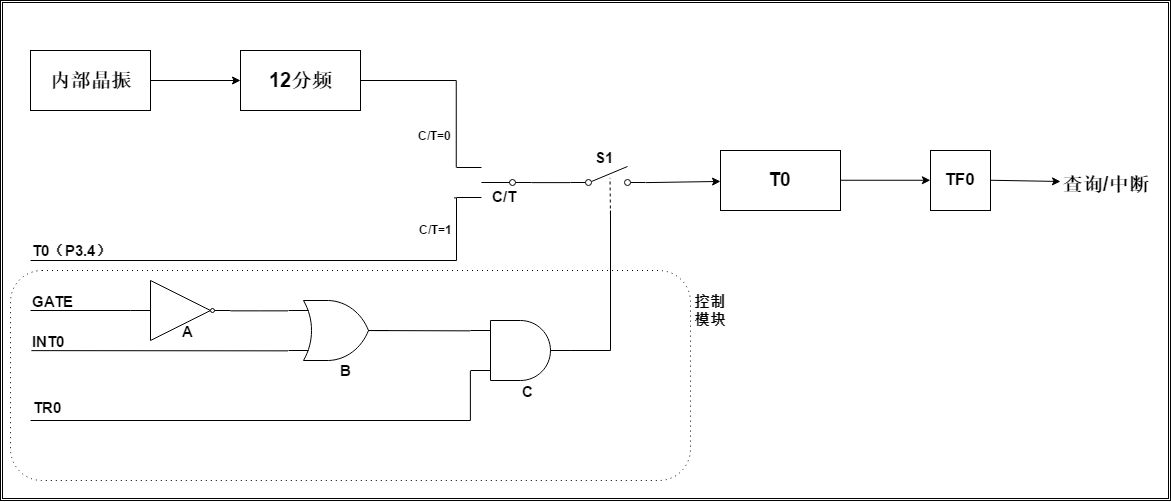
**思考 3**：当我们未设置的时候，TMOD默认设置为哪种工作方式，定时器还是计数器？  ****

图6- 4结构图(以T0为例)

**回答 3**：当单片机复位时TMOD=0，即当打开电源后默认设置为：Ｔ0和Ｔ1均为定时器方式０。

**6.2.2 TCON寄存器**

**TCON**控制寄存器(Timer/Counter Control Register) 容量为一个字节，字节地址8８H可按位寻址。因第5章已有详细介绍这里不再赘述。TMOD定义格式如表6-2。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TF1** | **TR1** | **TF0** | **TR0** | **IE1** | **IT1** | **IE0** | **IT0** |
| **8FH** | **8EH** | **8DH** | **8CH** | **8BH** | **8AH** | **89H** | **88H** |
| **（以定时器T0为例）**  **TF0:定时器T0溢出的标志位**。  当计时结束或计数溢出时，硬件自动将TF0位置1同时向CPU申请中断，当进入中断程序时硬件自动将TF0清0，TF0也可软件清0.  **TR0:定时器T0启停控制位**。  见图6-4，属于控制模块的一部分，和INT0，GATE共同控制定时器T0的启停 | | | | | | | |

表6- 2

我们主要需要TR0和TR1在GATE的协同下来控制定时器T0和T1 的启停。

**6.3 定时/计数器的工作方式（T0为例）**

**6.3.1 方式1**

当处于工作方式1时是一个16位定时/计数器，这16位由高8位TH0和低8位TL0这两部分组成，可以看作一个16位二进制的加1计数器，因此当累积数值超过则会产生溢出这时由硬件自动触发TF0置位向CPU申请中断，定时或计数完成，与此同时16位计数器清0.

**提示**：如果需要实现超过定时器硬件容量的计时可以结合软件来实现，在产生溢出时由硬件自动触发TF0置位向CPU申请中断通过软件重新装载初值，当符合软件设置的定时或计数条件时结束程序，从而实现长时间的定时功能。

下图为方式1 的逻辑结构图

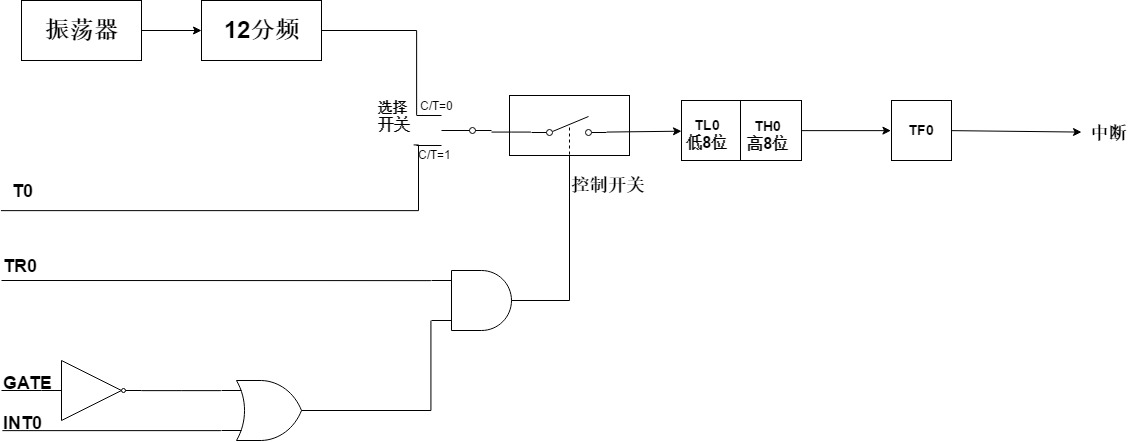


图6- 5方式1 的逻辑结构图

如果设置方式1需要将TMOD的M1M0设置为01，当选择开关C/T=0时为定时器以内置振荡器的12分频为脉冲信号，这时为定时器而C/T=1为计数器以外部T0（P3.4）为脉冲信号输入端。同时若使方式1 工作必须使控制开关为1，这里的“控制开关”由图6-4中的控制模块决定，若GATE=0定时器T0的启停只与TR0有关，反之定时器T0的启停由TR0和INT0共同决定。

因为方式1 的加1计数器容量为16位因此

定时时长：

机器周期:

a为需要设定的计数初值，为系统晶振（12MHz），因此当a=0时取得方式1最大的定时时长65536,即定时范围是0-65536.

**提示**：通过定时时长来确认需要装载的初值，这个初值需要进行16进制转换后高位初始化TH0，低位初始化为TL0，通过例6.1可以帮助你更加深入了解定时/计数器的使用。

**例6.1**单片机内部晶振频率为12MHz(12分频)，采用T0定时方式1控制LED以1ms为周期闪烁由P1.0口输出，并在虚拟示波器输出波形。

**解题步骤**：

1. 单片机为12分频，内部晶振频率为12MHz所以机器周期.
2. 因为要求信号输出周期是1ms所以定时器时长需为周期的一半即t=500,当定时满溢时将输出信号反转。不断重复这一过程即可实现输出以1ms为周期的脉冲信号.
3. 题目要求以方式1控制，方式1为16位定时/计数器，计数范围为.
4. 由机器周期和定时时长可得计数初值

⑤ 将a由10进制数转换为16进制数为0xfe0c.即需要设置T0定时器高8位TH0=0xfe低8位TL0=0x0c.

⑥编程要求定时器T0方式1即设置使用TMOD的低4位，因为题目只要求输出不受外部控制因此GATA=0，选择定时器方式设置C/T=0，选择方式1设置 M1M0=01，高4位无需设置即全初始化为0所以TMOD=0x01.

**参考电路图：**

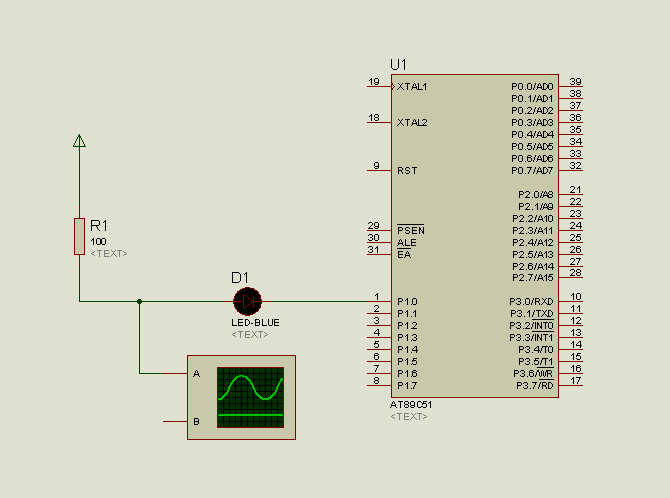


图6- 6

**C语言参考程序：**

1. **查询方式**

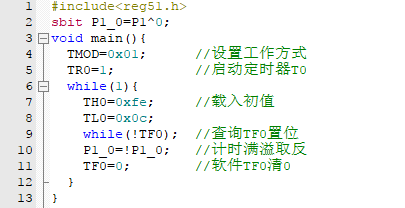


图6- 7

1. **中断方式**

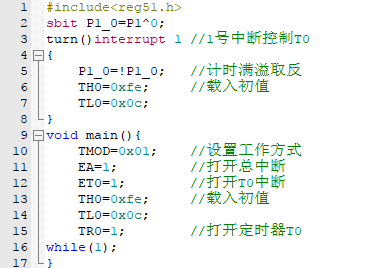


图6- 8

**运行结果：**

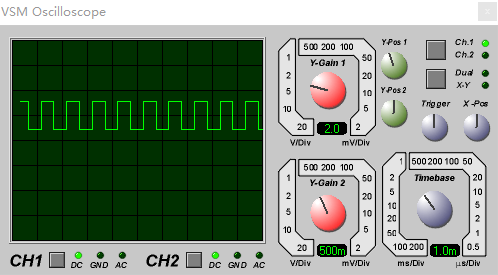


图6- 9

**思考5：**查询方式和中断方式有什么区别?两种方法有什么共同之处?

**回答5：**查询方式是指在一个死循环内不断扫描TF0的状态即查询符合定时/计数结束的条件，一旦通过说明计时结束随后将TF0复位；中断方式是通过硬件自动检测TF0并且复位的。两种方式都需要在计时结束后对初值进行初始化。

**例6.2** 单片机内部晶振频率为12MHz(12分频)，采用T0计数方式1测量从P3.4口输入的脉冲（150Hz），每测量100个控制P1.0口的LED闪烁一次。

**解题步骤**：

1. 题目要求以方式1控制，方式1为16位定时/计数器，计数范围为。
2. 由题目可知每测量100个脉冲产生中断控制P1.0的LED灯闪烁，因此计数值。

将a由10进制数转换为16进制数为0xff9c。即需要设置T0定时器高8位TH0=0xff低8位TL0=0x9c。

1. 编程要求定时器T0方式1只设置TMOD的低4位，题目只要求由T0来计数无外部控制因此GATA=0，选择计数器器方式设置C/T=1，选择方式1设置 M1M0=01，高4位无需设置即全初始化为0所以TMOD=0x05.
2. 脉冲生成方法设置

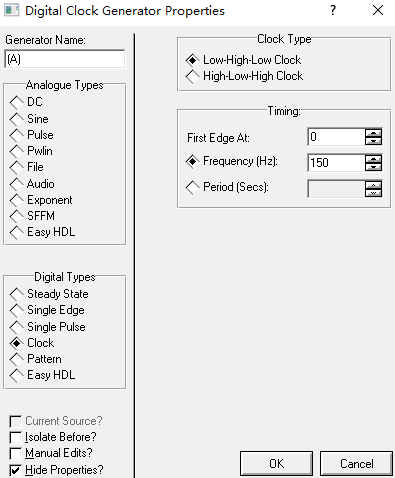


图6- 10

**参考电路图**

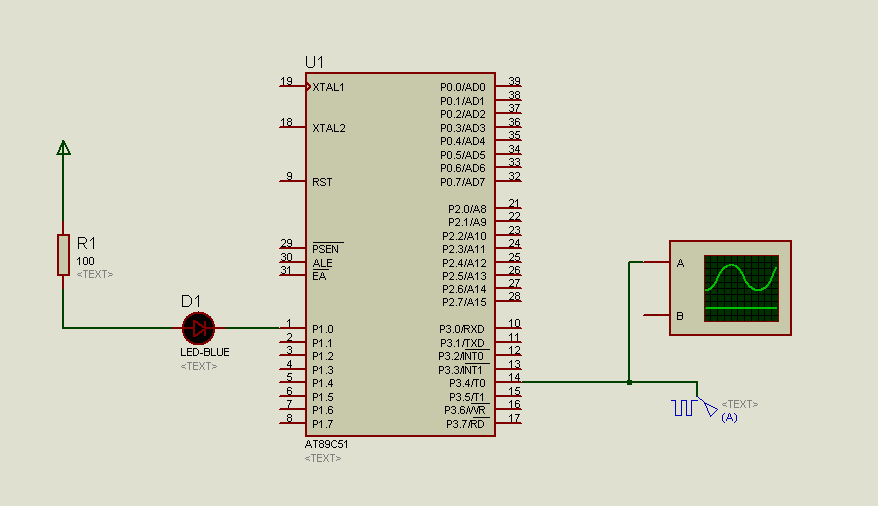


图6- 11

**C语言参考程序**

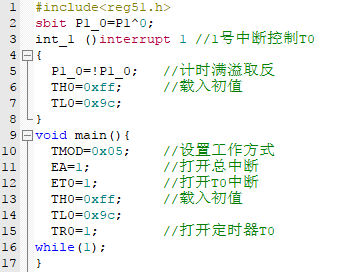


图6- 12

**6.3.2 方式2**

当处于工作方式2时是一个8位定时/计数器，这时会将方式1中的低8位TL0 作为二进制的加1计数器当累积数值超过则会产生溢出，与方式1不同的是高8位TH0用于存储一个8位的二进制初值当TL0溢出后会自动将TH0中的初值装入TL0中循环往复直至符合软件设置的停止条件为止，因此无需每次计时结束在软件中进行初始化，可以大大提高计时的精度，方式2是这4种方式中唯一能够自动重装的工作方式。尤其适合作为串行口波特率发生器。

下图6-14为方式2的逻辑结构图

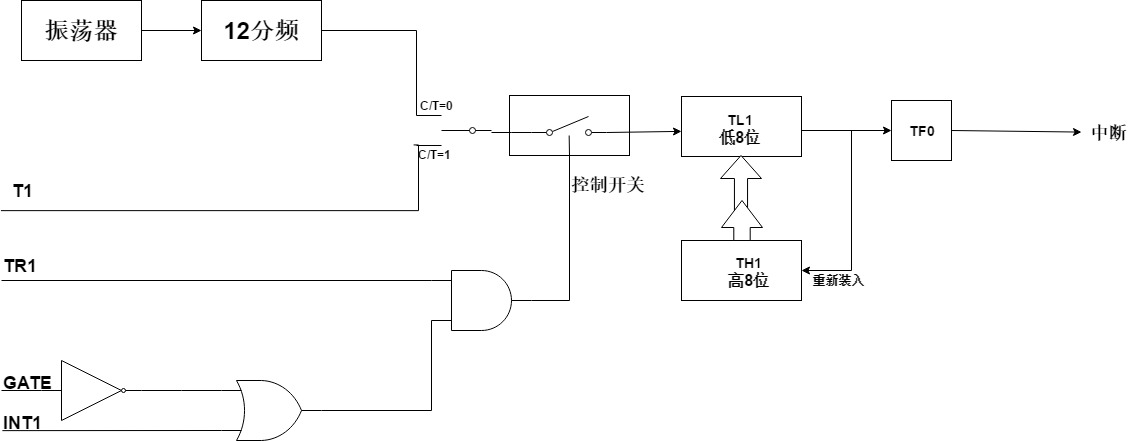


图6- 13方式2逻辑结构图

设置方式2与方式1类似，只需要将TMOD的M1M0置为10，其余基本不变，请参照方式1设置。

因为方式2 的加1计数器容量为8位因此

定时时长：

机器周期:

当a=0时取得方式1最大的定时时长256,即定时范围是0-256。

**例6.3**单片机内部晶振频率为12MHz，采用T0定时方式2控制LED以0.5ms为周期闪烁由P1.0口输出，并在虚拟示波器输出波形。

**解题步骤**：

1. 单片机为12分频，内部晶振频率为12MHz所以机器周期.
2. 因为要求信号输出周期是0.5ms，所以定时器时长需为周期的一半即t=250,当定时满溢时将输出信号反转。不断重复这一过程即可实现输出以0.5ms为周期的脉冲信号.
3. 题目要求以方式2控制，方式2为8位定时/计数器，计数范围为.
4. 由机器周期和定时时长可得计数值值。
5. 将a由10进制数转换为16进制数为0x06.因为方式2为8位自动重装定时器，当TL0计数溢出后自动将TH0的初值自动载入TL0中计数，不断重复这一过程直到命令终止，因此设置TL0=TH0=0x06。
6. 编程要求定时器T0方式2只设置TMOD的低4位，因为题目只要求输出不受外部控制因此GATA=0，选择定时器方式设置C/T=0，选择方式2设置 M1M0=10，高4位无需设置即全初始化为0所以TMOD=0x02.

参考电路图与方式1相同

**C语言参考程序**

1. **查询方式**

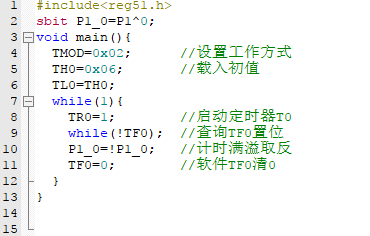


图6- 14

1. **中断方式**

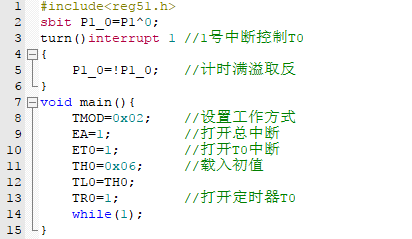


图6- 15

**结果分析**：通过方式2和方式1 的程序对比我们可以看出方式2的初值都只是在程序开始时候一次性初始化，运行过程没有装载是由硬件自动初始化，因此方式2不仅精确更简化了编程。

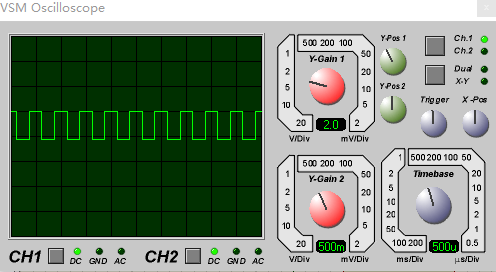
**运行结果：**

图6- 16

**例6.4**单片机内部晶振频率为12MHz(12分频)，采用T1计数器方式2测量例6.1中生成的脉冲，每计数100个控制P0.0口的LED灯闪烁

**解题步骤**：

①题目要求以方式2控制，方式2为8位定时/计数器，计数范围为.

②由题目可知每测量5个脉冲产生中断控制P0.0的LED灯闪烁，因此计数值。

③将a由10进制数转换为16进制数为0x9c。因为方式2为8位自动重装定时器，当TL1计数溢出后自动将TH1的初值自动载入TL1中计数，不断重复这一过程直到命令终止，因此设置TL1=TH1=0x9c。

④编程要求定时器T1方式2来测量例6.1生成的脉冲因此设置TMOD的低4位参考例6.1中的相关设置，设置TMOD的高4位时的题目只要求由T1来计数无外部控制因此GATA=1，选择计数器器方式设置C/T=1，选择方式2设置 M1M0=10，所以TMOD=0x61.

**参考电路图**

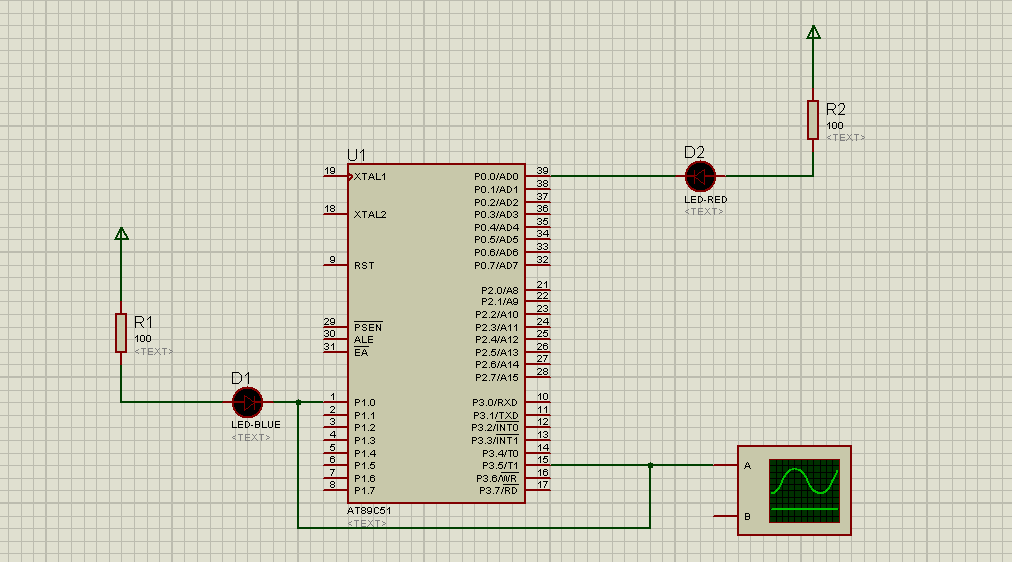


图6- 17

**C语言参考程序**



图6- 18

**提示**：如果你按照例题例题去做可能发现结果并不明显，因为例6.1中要求的是1ms为周期而每计数100控制P0.0闪烁只是将周期改为0.1s了这种区别用肉眼是难以发现的，你可以去掉代码中的注释部分，这样可以更加容易分析结果并且思考为什么。

**6.3.3 方式0**

当处于工作方式0时是一个13位定时/计数器，这13位由高8位TH0和低5位TL0这两部分组成，与方式1类似可以看作一个13位二进制的加1计数器，因此当累积数值超过则会产生溢出这时由硬件自动触发TF0置位向CPU申请中断，定时或计数完成，与此同时13位计数器清0.

我们可以看出，方式0除了与方式1的位数不同并无太大差别，因此设置方式请参考方式1。

下图为方式0 的逻辑结构图

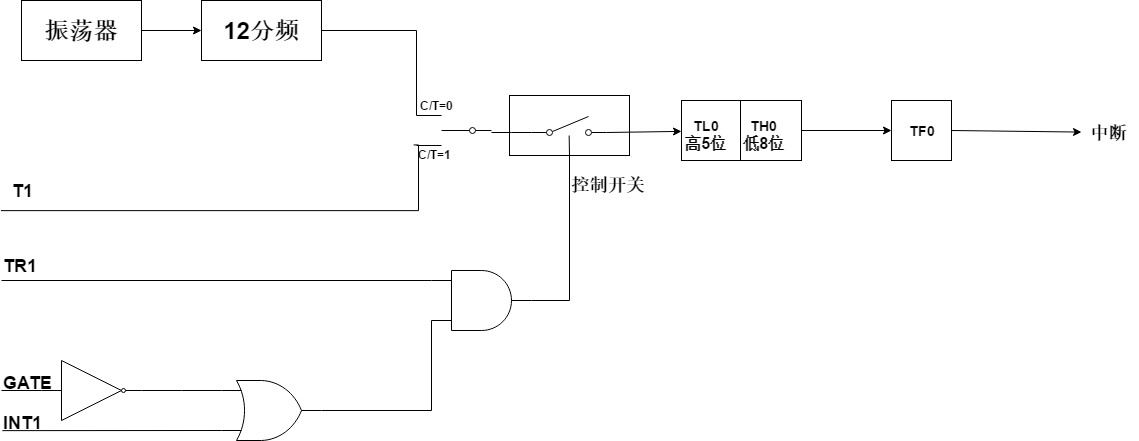


图6- 19方式0 逻辑结构图

设置方式0需要将TMOD的M1M0置为10，其余参考方式1。

因为方式0的加1计数器容量为13位因此

定时时长：

机器周期:

当a=0时取得方式1最大的定时时长8192μs,即定时范围是0-8192。

**提示**：需要注意的是方式0由于选取位特殊计算初值应按照从大到小填写，而16位寄存器的5-6位插入‘000’计算例如例6.1用方式0实现的话

B

调整后

因为方式0是为了兼容早期单片机MCS-48系列产生的，计算比较复杂因此 在实际应用时往往用方式1来代替方式0.

**例6.5**单片机内部晶振频率为12MHz(12分频)，采用T0定时方式0控制LED以1ms为周期闪烁由P1.0口输出，并在虚拟示波器输出波形。

**解题步骤**：

1. 单片机为12分频，内部晶振频率为12MHz所以机器周期.
2. 因为要求信号输出周期是1ms所以定时器时长需为周期的一半即t=500,当定时满溢时将输出信号反转。不断重复这一过程即可实现输出以1ms为周期的脉冲信号.
3. 上面已经计算了初值a=

④ 编程要求定时器T0方式0即设置使用TMOD的低4位，因为题目只要求输出不受外部控制因此GATA=0，选择定时器方式设置C/T=0，选择方式0设置 M1M0=00，高4位无需设置即全初始化为0所以TMOD=0x00.

**参考电路图：**参考图6-6

**C语言参考程序：**

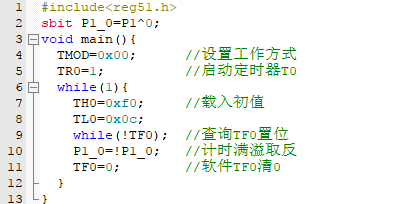


图6- 20

**6.3.4 方式3**

方式3比较复杂，能够实现的功能也比较丰富。通常方式3有三种组合方式①TH0+TF1+TR1实现带中断8位定时器。②TL0+TF0+TR0实现带中断8位定时/计数器③T1无中断定时/计数器。下图6-16为方式3的逻辑结构图.

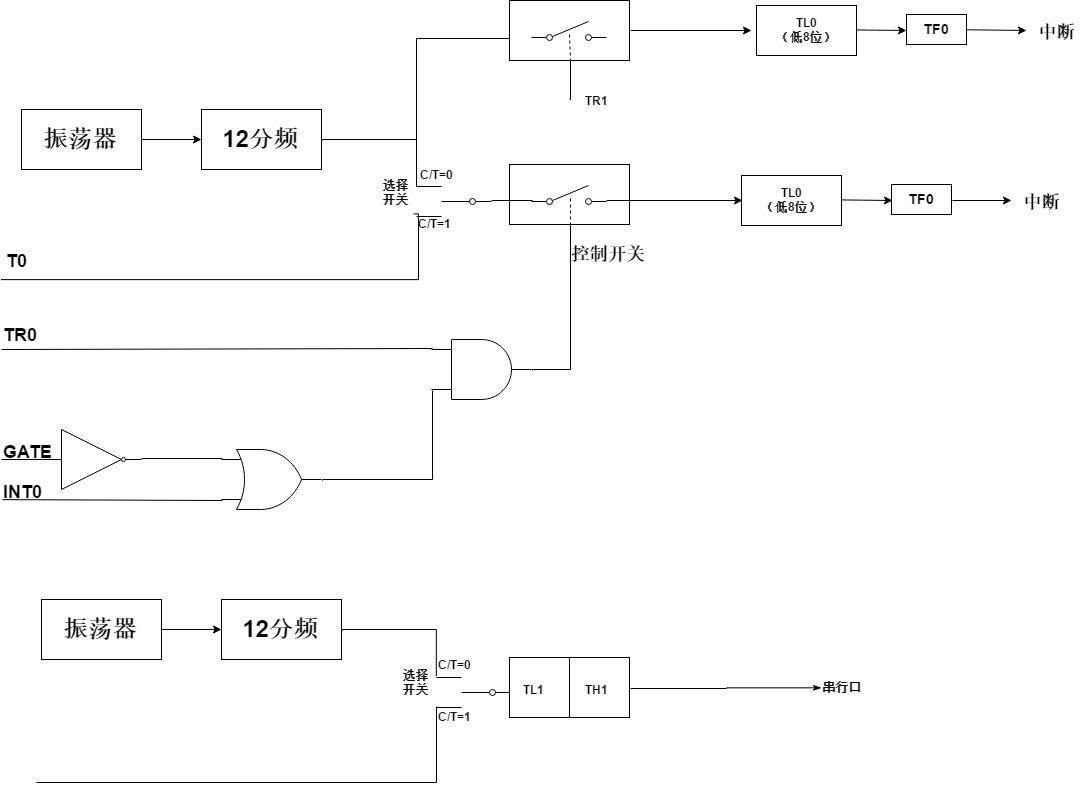


图6- 21方式3逻辑结构图

如果设置方式3需要将TMOD的M1M0置为00。工作方式3下只有定时器T0可以工作而定时器T1没有方式3。当定时器T0设置为方式3时，将原定时器/计数器T0分为TL0和TH1两个8位独立的加一计数器，其中TL0既可以用于定时器方式也可以作为计数器方式，而TH0则只能用于定时器方式。如果强行设置T1为工作方式3，T1将停止工作。

通过结构图我们可以看出工作方式3下，TL0沿用了原定时器T0的控制位，引脚和中断源。既可以用于定时器方式也可以作为计数器方式，具体设置和功能同方式1，区别在于这时是8位加1计数器而不是16位。TH0则只能用于定时器方式，作为8位定时器使用，因为它占用了定时器T1的控制位TR1和标志位TF1，所以启停只受TR1的控制。在定时器T0在方式3工作时T1仍然可以在方式1，方式2和方式0工作。

**6.4 定时/计数器的应用**

在我们的日常生活和工业生产中，单片机的定时/计数器应用十分广泛因为其小巧灵活、成本低、易于产品化。能方便地组装易于编程和控制的特点备受欢迎。也有通过编程作为分频器来产生不同频率的方波来使用的案例，总之，通过灵活运用单片机的定时/计数器功能可以解决许多现实生活的问题，本小节将详细介绍定时/计数器在生活中的应用。

**例6.6**

图6- 22

**情景引入：**生活中处处可见五颜六色的霓虹灯用流水灯显示的招牌，圣诞节的彩灯圣诞树…

**设计要求：**用6个不同颜色的LED灯，四个控制开关来模拟彩灯。①正常情况6灯交替闪烁②按下开关1为灯1→灯6的流水灯,默认频率为1s③按下开关1为灯6→灯1的流水,灯默认频率为1s④开关3具有将流水灯闪烁频率加倍的效果⑤开关4则可以将流水灯切换回彩灯的交替闪烁。⑤采用方式1实现。

**思路点拨：**

1. 通读题目可以看出我们需要实现两个基本功能，彩灯的交替闪烁和能够改变频率的流水灯。
2. 彩灯闪烁没有周期限制要求因此比较简单，只需要简单的赋值和利用键盘扫描的延时即可实现。
3. 流水灯需要4个开关来控制,我们可以通过上一章的中断来实现。
4. 对延时函数的设计为本题的重点，通过题目我们可以知道我们需要设计一种能够自由改变定时时长的延时函数，因此需要传递一个参数来控制定时的时长。那么我们可以先在delay（）中内置一个“原子”定时器，传递进来的参数为这个定时时长的倍数那么不就可以实现任意时长了么。具体实现请参考图6-22.

**参考电路图**

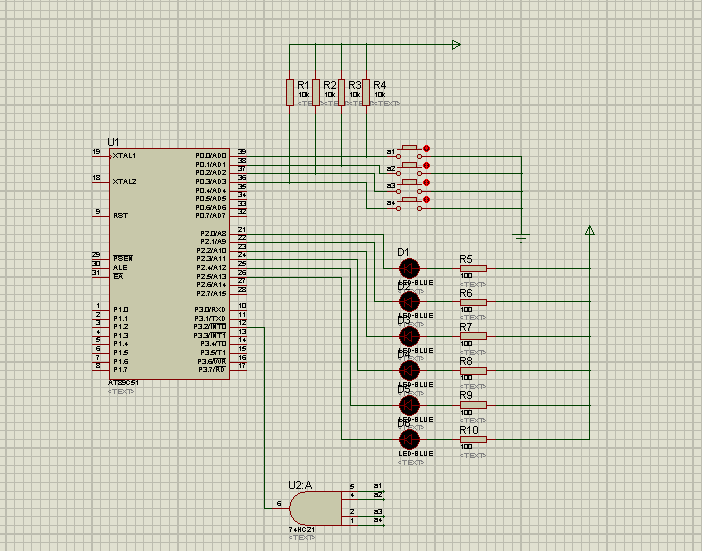


图6- 23

**C语言参考程序**

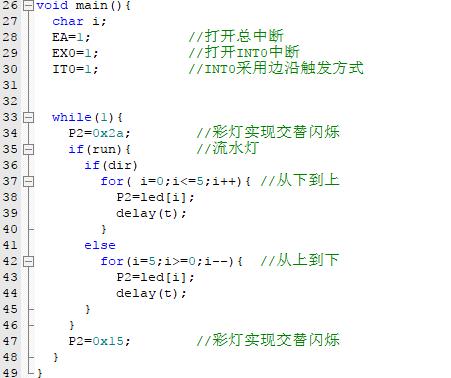
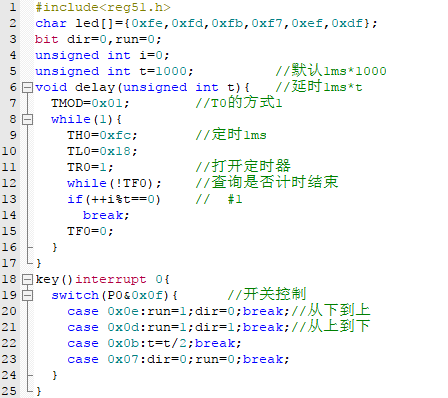


图6- 24

**思考**：为什么不要求在交替闪烁时加入周期控制而只在流水灯加入周期控制？

**回答**：因为要求开关控制需要时刻对开关状态进行扫描，这部分扫描是需要时间的因此不可能做到精确延时的交替闪烁，但我们可以利用扫描的延时来实现彩灯的交替闪烁。并且生活中类似流水灯的商业广告比较常见我们强调的是这一方面的应用。

**例 6.7**

**情景引入：**随着社会的发展，人们越来越关心自身是身体健康，运动已经成为我们生活中必不可少的部分。在健身房里的动感单车是一种非常棒的有氧运动，能有效提高心、肺功能，力量和耐力。消耗能量、降低体脂同时还有缓解心理压力的作用…

图6- 25

**设计要求：**假设单车车轮每转动10周消耗3kJ的热量，每天需要消耗1560KJ的热量①在P3.5口输入一个周期为5Hz的脉冲信号，模拟车轮的转动频率并且连接LED灯显示②通过对车轮转动频率的测算来计算运动所消耗的热量并在4位LED数码管显示今天待消耗的热量。③设有重置开关用来重置运动量。④采用方式2实现。

**思路点拨：**

①本题为7位LED数码管和计数器的简单应用。

②根据题目要求我们可以设置一个方式2的计数器测量外部脉冲的频率来模拟对车轮转动圈数对计数，因此初值设定为246。

③题目说明每天对需要消耗对热量为1560kJ我们可以设置为全局变量Q，每当计数器溢出触发中断，在中断中对Q进行修改。

④重置开关我们可以通过外部中断实现。

**参考电路图：**

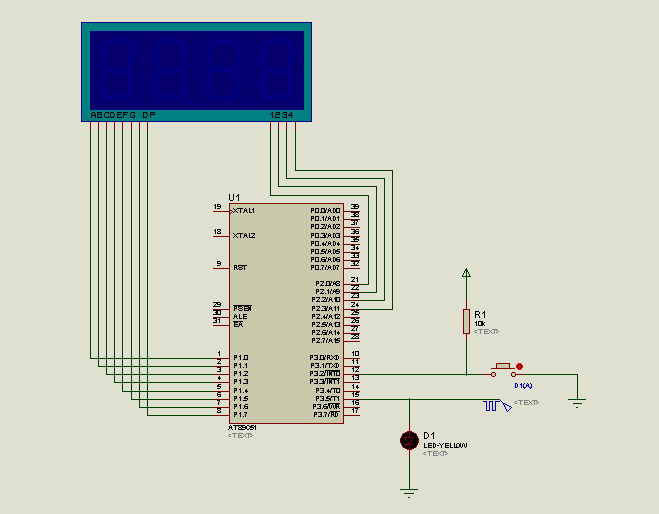


图6- 26

**C语言参考程序：**

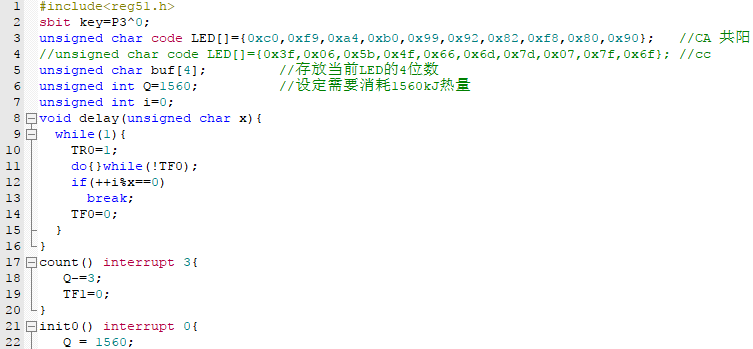
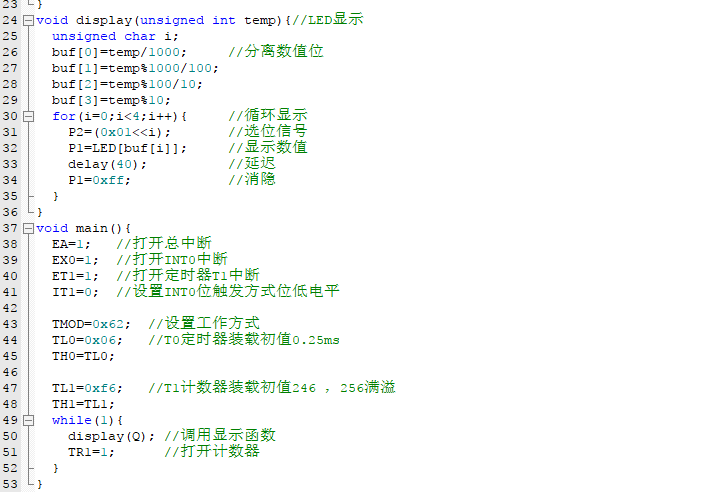
 

图6- 27

**运行结果：**

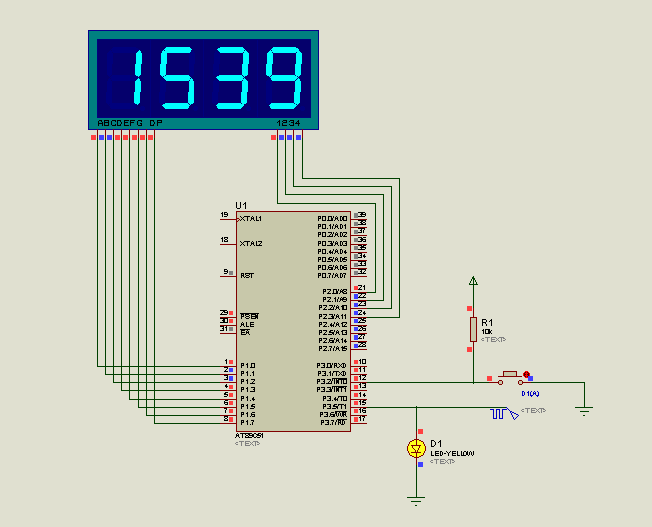


图6- 28

**提示：**本题需要对7段数码管有一定对了解，了解运行对原理和0-9对应的数码值，并且画电路图时注意区分共阳（CA）和共阴数码管（CC）不同数码管应选取不同的编码。

**例 6.8**

图6- 29

**情景引入：**随着发令枪的一声枪响，运动员争先恐后的跑起来，同时裁判员沉着镇定的按下了秒表…

**设计要求：**1个4位数码管显示计数值，精确到0.01s 最长计时100s 。设置2个开关①打开开关1开始计时，关闭开关暂停②按下开关2将当前数码管显示值重置为0③采用计数器T1的方式2实现。

**思路点拨：**

①本题通过计数器来模拟实现来秒表的部分功能。

②

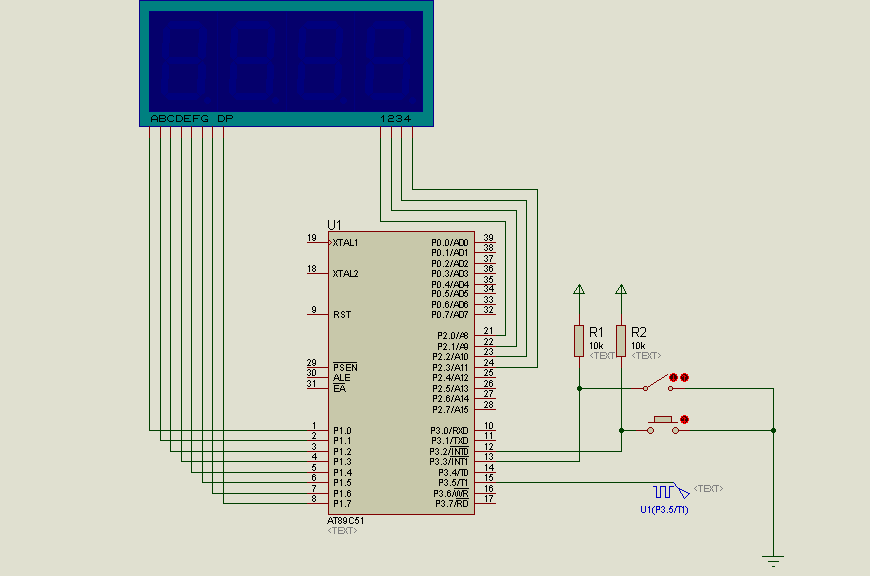
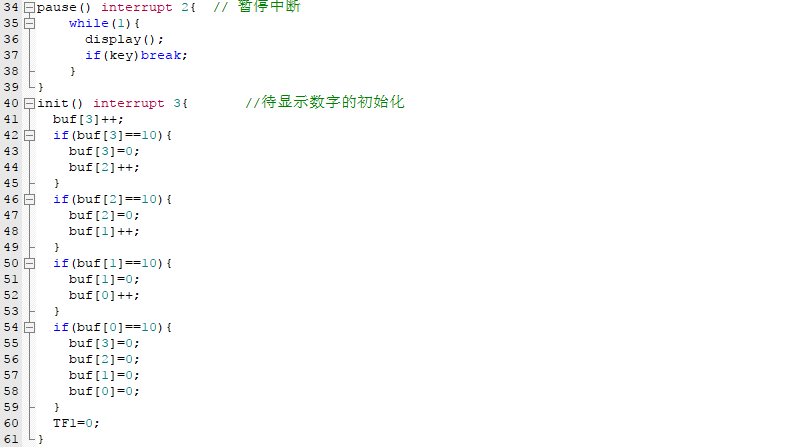


图6- 30





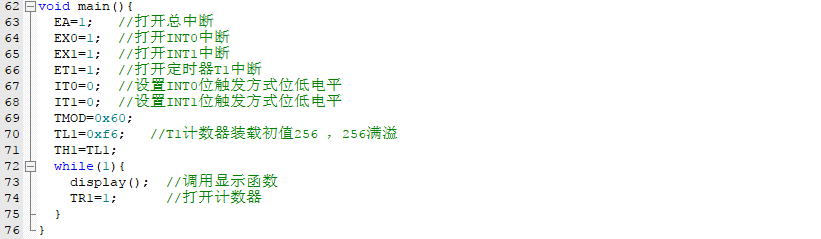


图6- 31

**运行结果：**

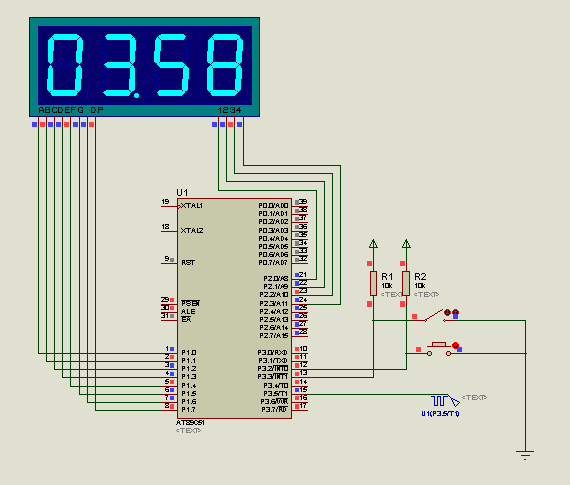


图6- 32

**本章小结**

1.51单片机定时/计数器的基本原理时利用加1计数器和系统内部晶振来实现定时和计数的。本质上来说都是计数器的应用。

2定时/计数器通过中断的溢出来引起CPU的响应，需要与外部中断区别，定时/计数器的中断是自动触发的。

3定时/计数器有4种工作方式，方式2可自动重装因此定时精确对定时要求高的场合可以优先考虑方式2；方式0为13位定时/计数器因为计算初值麻烦一般用方式1替代；方式3最为复杂，方式3的T1定时器不可用只能设置T0.

4.TMOD用来设置定时/计数器的0-3四种工作方式和对定时/计数模式的选择。

5.TCON用来设置定时/计数器的溢出标志位和请求方式位。

6. 定时/计数器一般的编程过程：

①确定定时或计数，设置工作方式（TMOD设置）

②计算初值，注意区分不同方式的设定范围。

③设置初值，高位和低位。

④如需中断，需要打开中断并且设置中断内部的功能。

**注意：**方式2的自动重装因此高位和低位相同。方式2无需重载初值但其他3种方式都涉及到初值的重载。

**课后习题**

1.51单片机定时/计数器的基本工作原理是什么?

2.定时/计数器有几种工作方式?这些工作方式有什么功能上的区别?

3.定时模式和计数模式怎么设置?有什么区别?

4.方式2与其他方式相比有什么区别?方式2 适用于什么场合?

5.设晶振ƒ=6MHz,要求分别使用方式1,方式2和方式0在P1.0口输出一组T=0.5ms的方波。

6.同题目5输出占空比60%的方波。

7.单片机的晶振ƒ=12MHz，通过编程使之对题目5产生的脉冲进行计数，每计数2000个控制一个LED小灯亮1s循环10个周期。

8.请将例6.8用定时器T0方式2来实现。

9. **情景引入：**时钟处处可见， 正是有了时钟，才可以让我们记录生活的点点滴滴。正是有了时钟，我们才可以把时间把握到分分秒秒…

图6- 33

**设计要求：**使用4位数码管来模拟简单的时钟。

10. **情景引入：**每个人都见过红绿灯，红绿灯让我们的交通变得井井有条，同时也避免了许多的交通事故，用一种简单的方式来处理复杂的交通…

**设计要求：**模拟十字路口的红绿灯，红灯20s，绿灯17s,黄灯3s

图6- 34