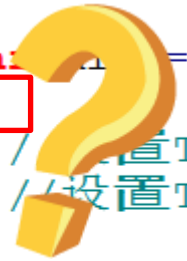
项目四：定时与中断系统应用

——简易秒表的设计（2）

任务引入

1.c

```
34 void main() //主函数
35 {
36     while(1)
37     {
38         unsigned cha  =0;      //秒计数器定义
39         TMOD=0x10;             //设置T1为工作方式1
40         TH1=0x3c;     //设置T1计数初值高8位，定时时间50ms
41         TL1=0xb0;     //设置T1计数初值低8位
42         TR1=1;                 //启动定时器开始计数
43         while(1)
44         {
45             disp(miao);              //显示秒计数器值
46             delay1s();               //调用1s函数
47             miao++;                  //秒计数器加1
48             if(miao==100)
49                 miao=0;       //秒计数计满，则从0开始计数
50         }
51     }
52 }
53
```



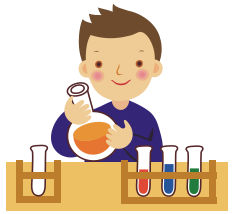
后续内容，调试程序用，先写上

教学目标

知识目标：

1、掌握定时器相关寄存器设置

2、掌握定时器的编程过程



能力分析：

1、能够灵活利用查询方式实现定时程序的编写

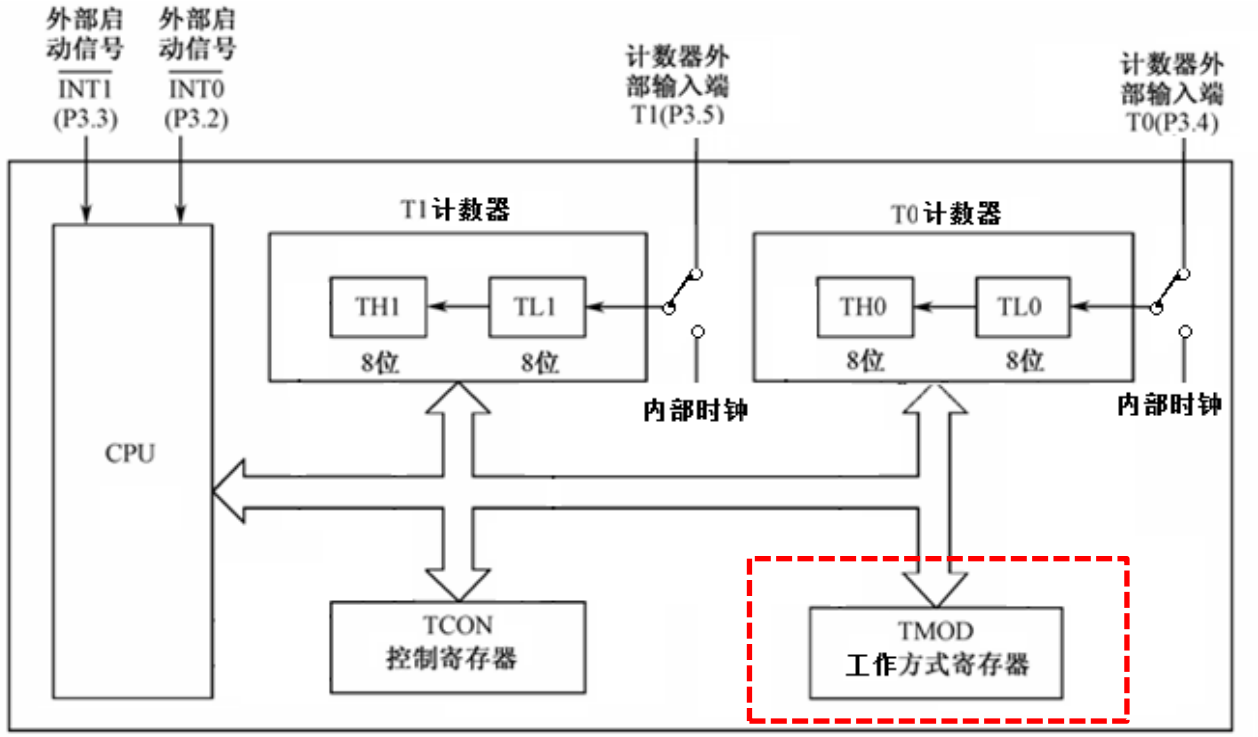
2、能够根据要求完成简易秒表程序的修改

素质目标：

1、培养独立自主的学习能力和团队协作精神

2、增强实践能力，培育工匠精神，提高解决实际问题的能力

TMOD是什么？



工作方式寄存器：

1.选择定时计数器的4种工作方式

2.选择定时or计数

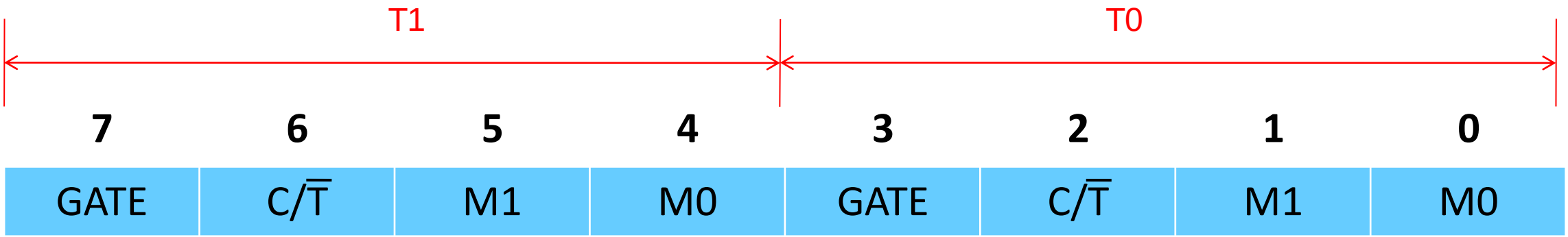
3.决定启动方式

TMOD

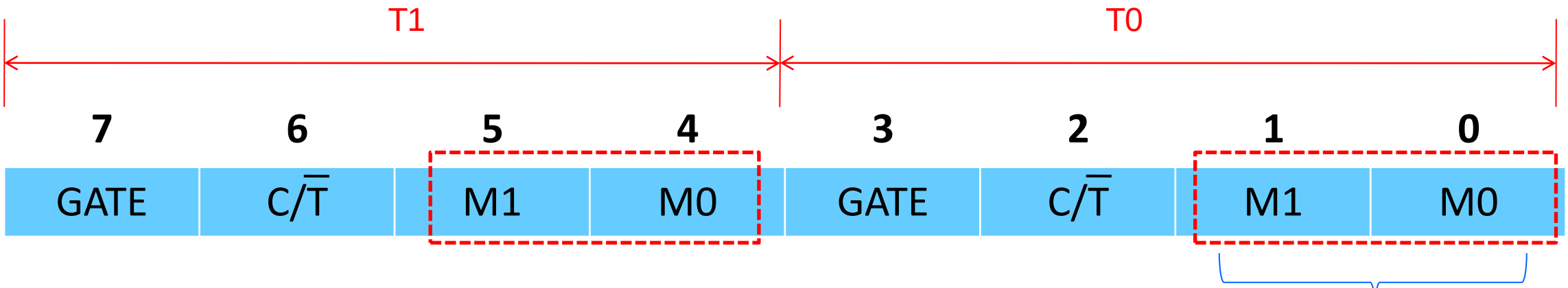
字节地址：89H，不可以位寻址。

SFR	字节地址
B	F0H
ACC	E0H
PSW	D0H
IP	B8H
P3	B0H
IE	A8H
P2	A0H
SBUF	99H
SCON	98H
P1	90H
TH1	8DH
TH0	8CH
TL1	8BH
TL0	8AH
TMOD	89H
TCON	88H

TMOD的格式



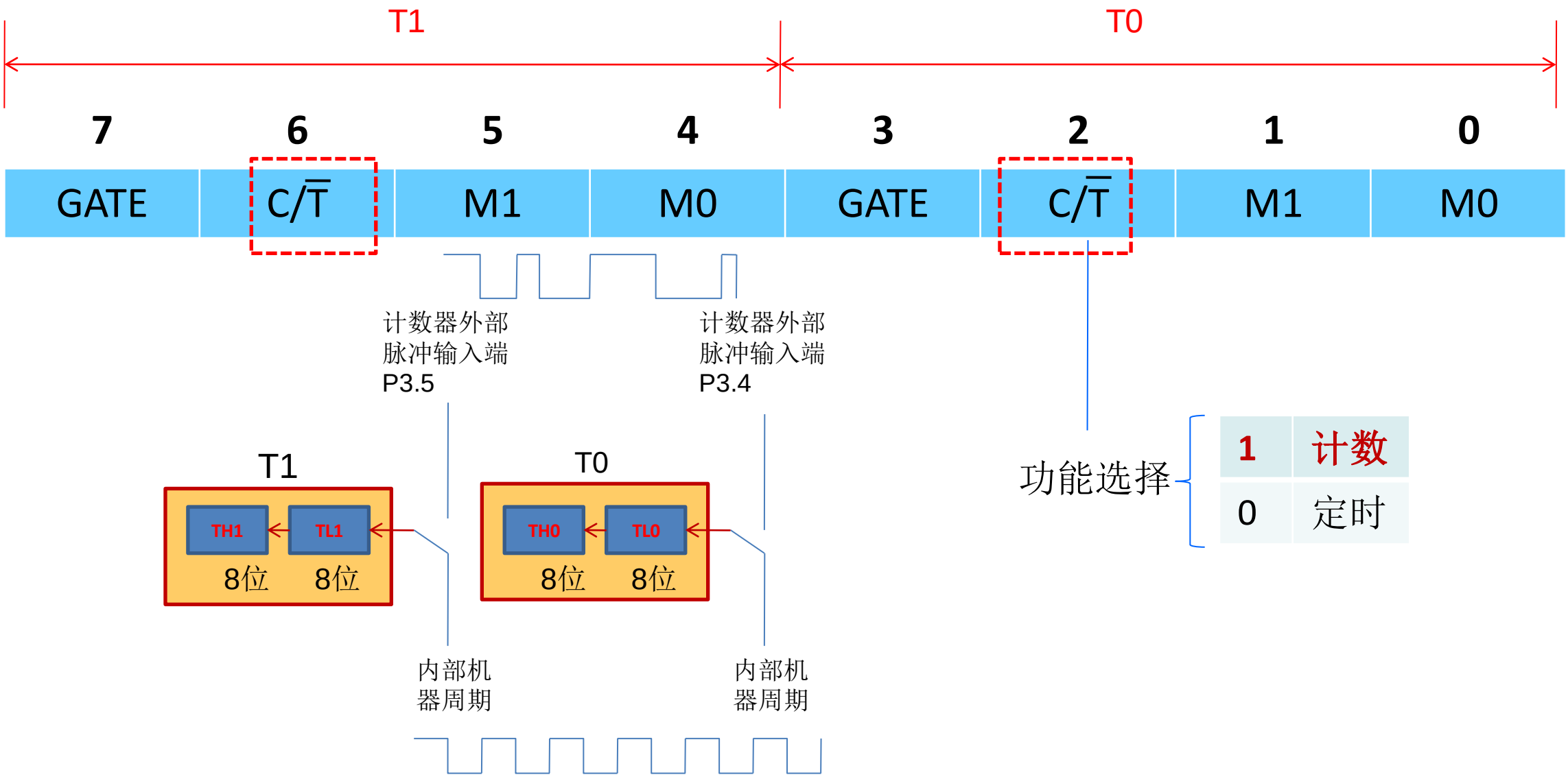
TMOD——工作方式



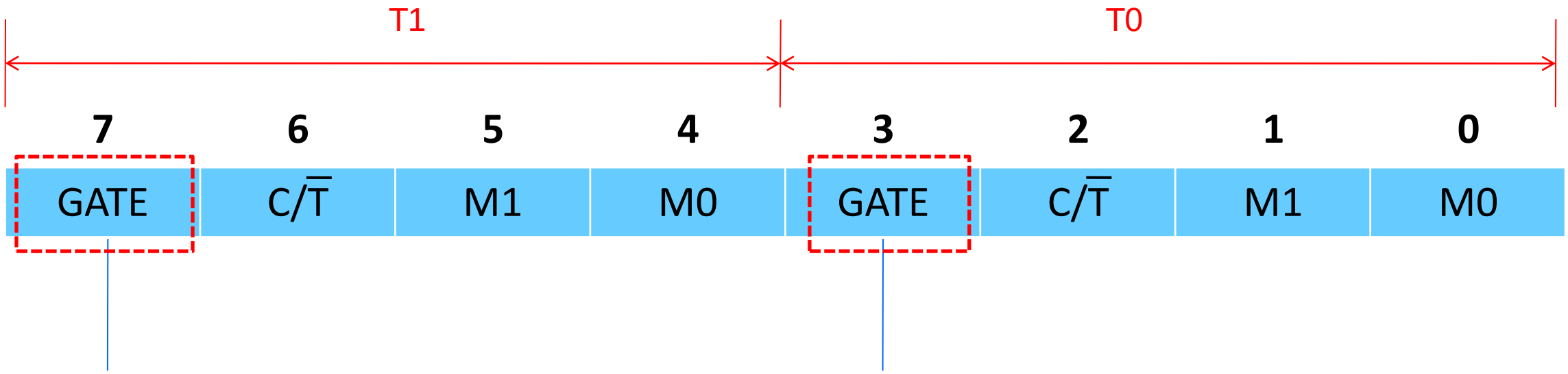
功能	工作方式	M1	M0
13位计数器	方式0	0	0
16位计数器	方式1	0	1
8位计数器，初值自动重装	方式2	1	0
分成两个8位计数器，仅用于T0。T1停止使用	方式3	1	1

工作方式

TMOD——功能选择


T1
T0
7
6
5
4
3
2
1
0
GATE
C/T
M1
M0
GATE
C/T
M1
M0
计数器外部
脉冲输入端
P3.5
计数器外部
脉冲输入端
P3.4
T1
TH1
TL1
8位
8位
T0
TH0
TL0
8位
8位
内部机
器周期
内部机
器周期
功能选择
1
计数
0
定时

TOMD——启动控制的选择



1	TR1=1，$\overline{INT1}$=1共同启动定时器
0	TR1=1启动定时器，不受$\overline{INT1}$控制

1	TR0=1，$\overline{INT0}$=1共同启动定时器
0	TR0=1启动定时器，不受$\overline{INT0}$控制

TOMD——具体编程应用

7	6	5	4	3	2	1	0
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
0	0	0	1	0	0	0	0

TMOD=0x10; 十六进制数10H(00010000)赋值给TMOD寄存器

1. 这是一个字节赋值的操作。
2. 代表什么含义呢？

TOMD——再举例巩固知识

设定时器T0为定时工作方式，要求用软件启动定时器T0工作，按方式1工作；定时器T1为计数工作方式，要求软件启动，工作方式为方式2。如何设置TMOD？

7	6	5	4	3	2	1	0
GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
0	1	1	0	0	0	0	1

TMOD=0x61;

TOMD——小工具深化理解

TMOD的设置					
T1			T0		
GATE	C/T	M1M0	GATE	C/T	M1M0
软件启动	定时器	工作方式2	软件启动	计数器	工作方式3
0	0	10	0	1	11

自主开发的
TMOD设置教
学小工具

修改程序

```
1.c
34 void main() //主函数
35 {
36     while(1)
37     {
38         unsigned char miao=0;      //秒计数器定义
39         TMOD=0x10;                 //设置T1为工作方式1
40         TH1=0x3c;    //设置T1计数初值高8位，定时时间50ms
41         TL1=0xb0;    //设置T1计数初值低8位
42         TR1=1;                     //启动定时器开始计数
43         while(1)
44         {
45             disp(miao);            //显示秒计数器值
46             delay1s();             //调用1s函数
47             miao++;                //秒计数器加1
48             if(miao==100)
49                 miao=0;      //秒计数计满，则从0开始计数
50         }
51     }
52 }
53
```

如果我们使用的是定时器T0，程序要如何修改？

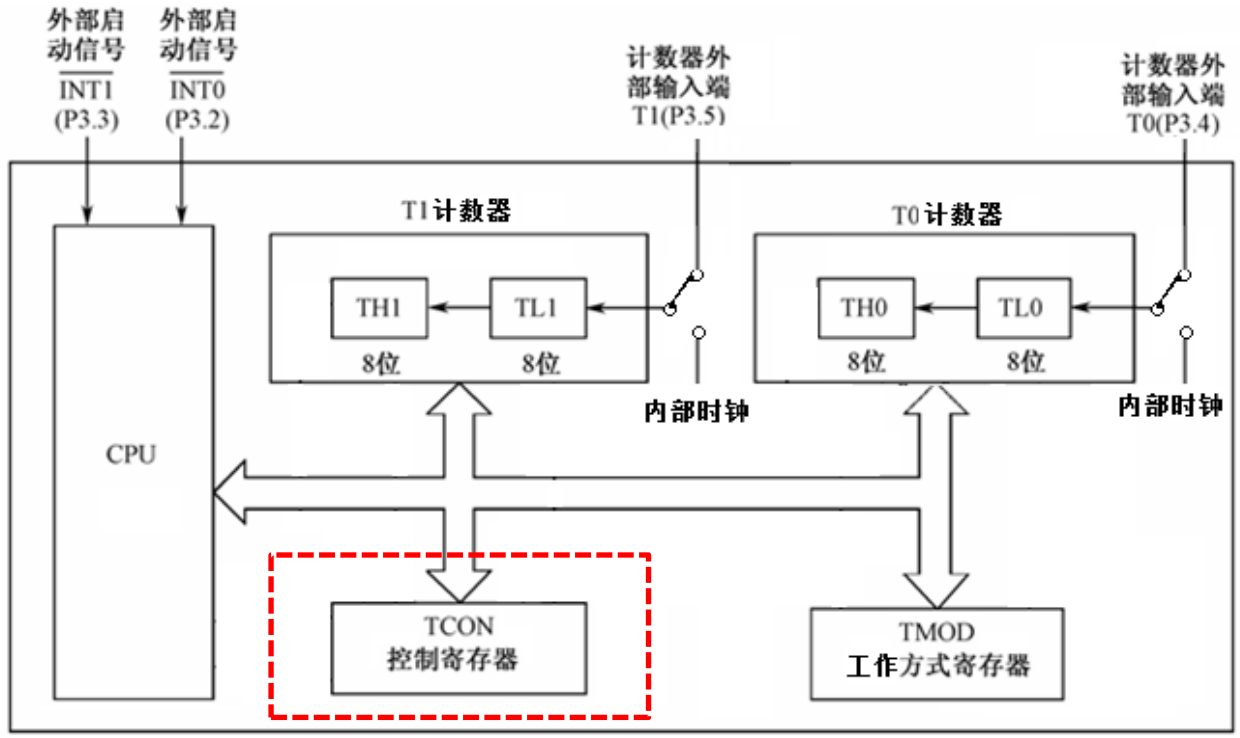
修改完TMOD值，观察运行现象，分析原因。

控制寄存器TCON

控制寄存器：

1.启动控制

2.计满溢出控制

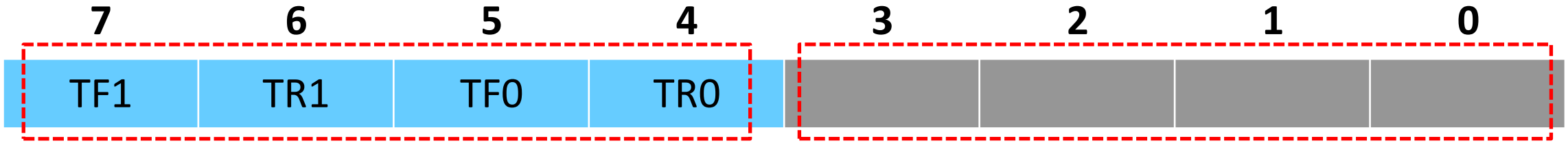


TCON

字节地址：0x88，可以位寻址。
系统复位时，所有位均清零。

SFR	字节地址
B	F0H
ACC	E0H
PSW	D0H
IP	B8H
P3	B0H
IE	A8H
P2	A0H
SBUF	99H
SCON	98H
P1	90H
TH1	8DH
TH0	8CH
TL1	8BH
TL0	8AH
TMOD	89H
TCON	88H

TCON的格式



高4位：
分别控制T0、T1的启动与溢出

低4位：
和定时计数器无关
中断系统中才使用

TCON——启动控制

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0				

T0启动位	TR0=1,启动T0工作
	TR0=0,停止T0工作

T1启动位	TR1=1,启动T1工作
	TR1=0,停止T1工作

1.c

```
13 //TH1=0x3c TL1=0xb0
14 void delay1s()   //1s延时函数
15 {
16     unsigned char i;
17     for(i=0;i<20;i++)
18     {
19         TH1=0x3c;
20         TL1=0xb0;
21         //启动定时器
22         TR1=1;
23         //如何处理溢出
24         while(!TF1);
25         TF1=0;
26     }
27 }
```

TR1=1；　// 位操作指令，启动T1

TCON——溢出标志

1	计数溢出，硬件自动置1
0	中断方式：硬件自动清0
	查询方式：软件清0

TF0：定时器T0的溢出标志位。
TF1：定时器T1的溢出标志位。

TCON——溢出标志的应用

取反

```
while(!TF1);      //查询TF1，这是查询方式的应用
TF1=0;            //如果计满溢出，将TF1清0
```

1.c

```
13 //TH1=0x3c TL1=0xb0
14 void delay1s()   //1s延时函数
15 {
16     unsigned char i;
17     for(i=0;i<20;i++)
18     {
19         TH1=0x3c;
20         TL1=0xb0;
21         //启动定时器
22         TR1=1;
23         //如何处理溢出
24         while(!TF1);
25         TF1=0;
26     }
27 }
```

TCON——中断方式下 TF的应用

何时置1?
如何清0?

举生活中例子说明两种处理方式的不同
中断方式后续深入讲解

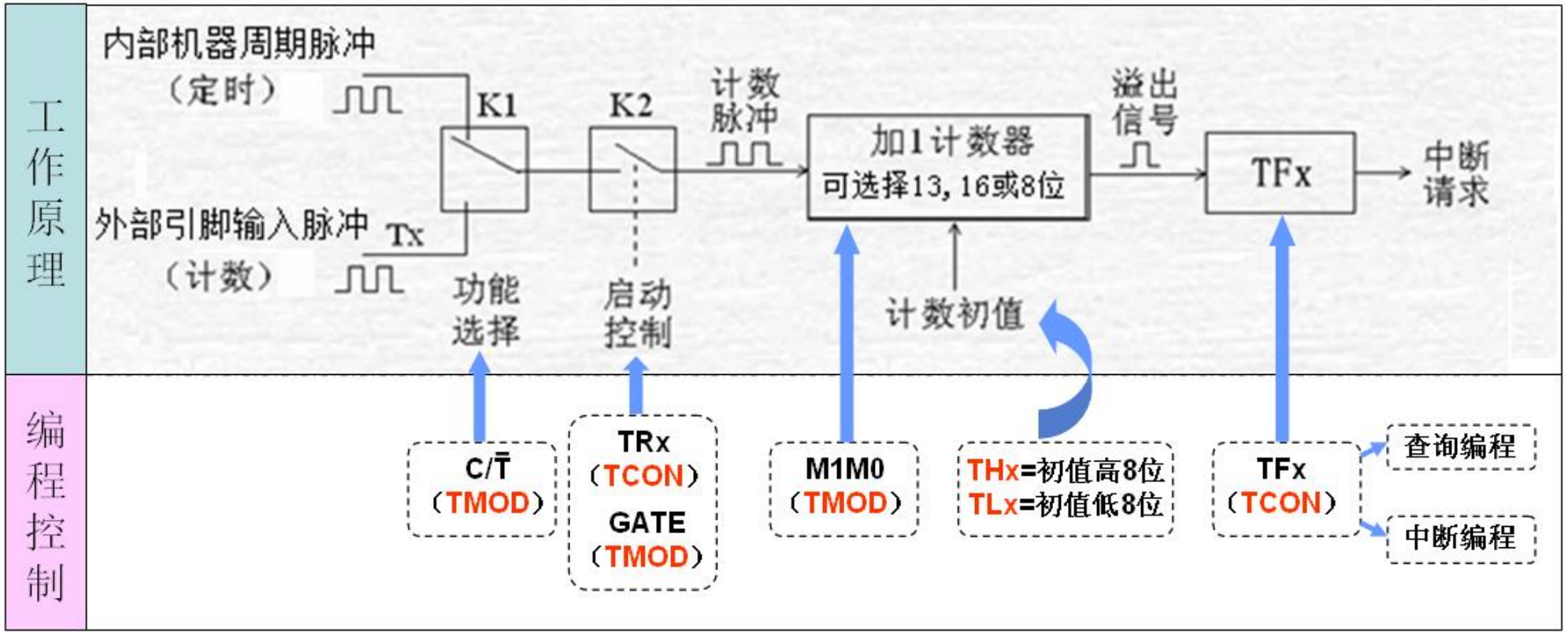
修改程序

1.c

```
13  //TH1=0x3c TL1=0xb0
14  void delay1s()   //1s延时函数
15  {
16      unsigned char i;
17      for(i=0;i<20;i++)
18      {
19          TH1=0x3c;
20          TL1=0xb0;
21          //启动定时器
22          TR1=1;
23          //如何处理溢出
24          while(!TF1);
25          TF1=0;
26      }
27  }
28  void disp(unsigned char i)//数码管显示函数
29  {
30      unsigned char led[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90} ;//0～9
31      P1=led[i/10];    //显示i高位
32      P2=led[i%10];    //显示i低位
33  }
34  void main() //主函数
35  {
36     while(1)
37     {
38          unsigned char miao=0;     //秒计数器定义
39          TMOD=0x10;                //设置T1为工作方式1
40          TH1=0x3c;    //设置T1计数初值高8位，定时时间50ms
41          TL1=0xb0;    //设置T1计数初值低8位
```

使用定时器T0，完成整个程序的修改。

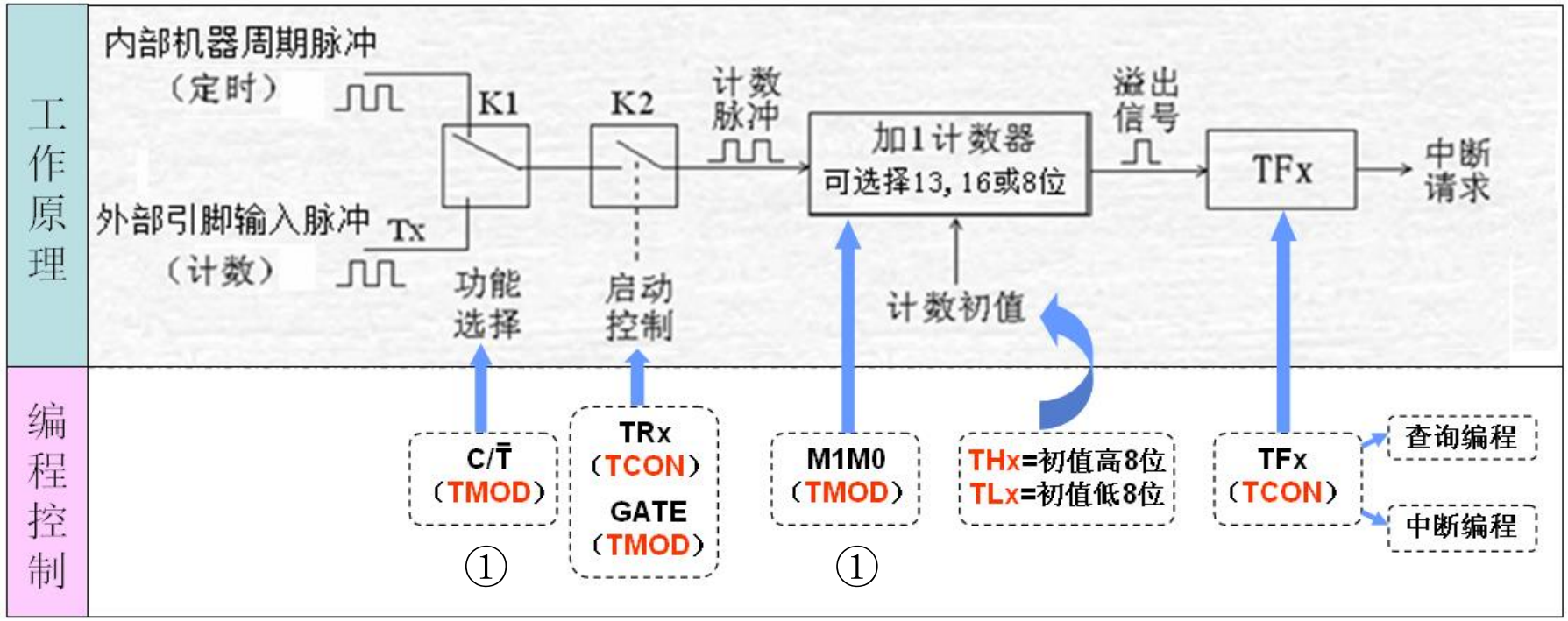
总结——定时器/计数器的编程控制过程



总结——定时器/计数器的编程控制过程

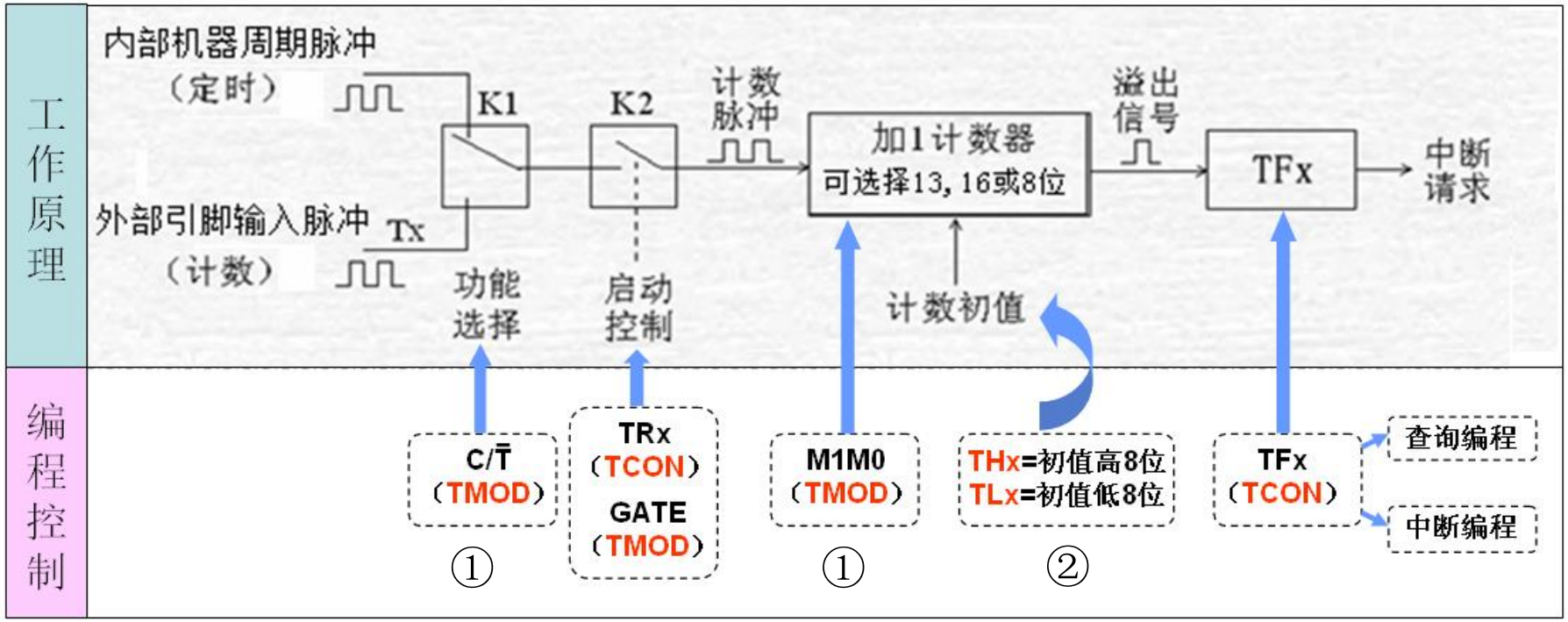
步骤一： 设置工作方式

写TMOD
设置工作方式



总结——定时器/计数器的编程控制过程

步骤二： 初始值设置

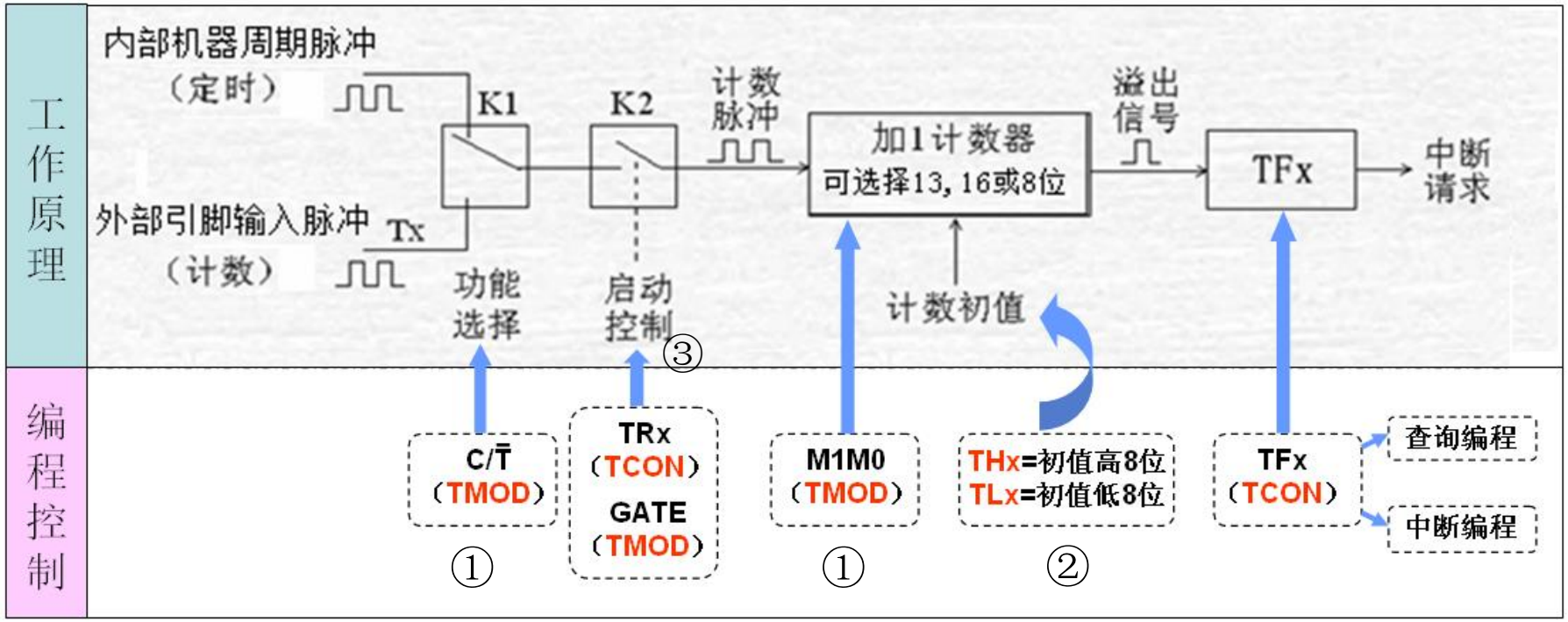


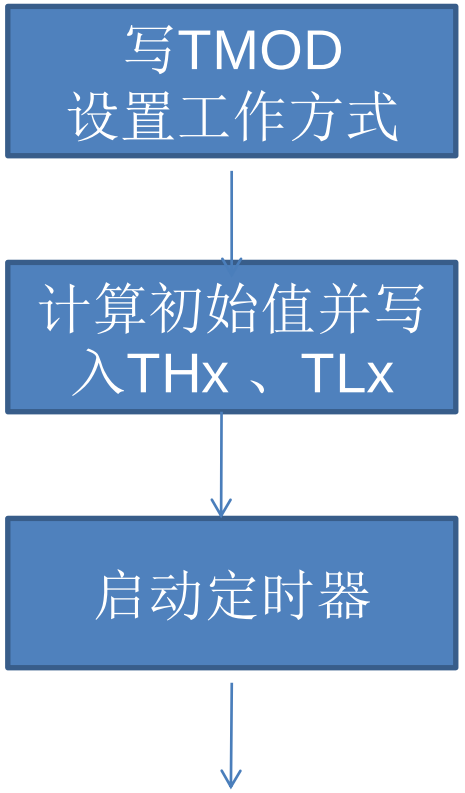
写TMOD
设置工作方式

计算初始值并写入THx 、TLx

总结——定时器/计数器的编程控制过程

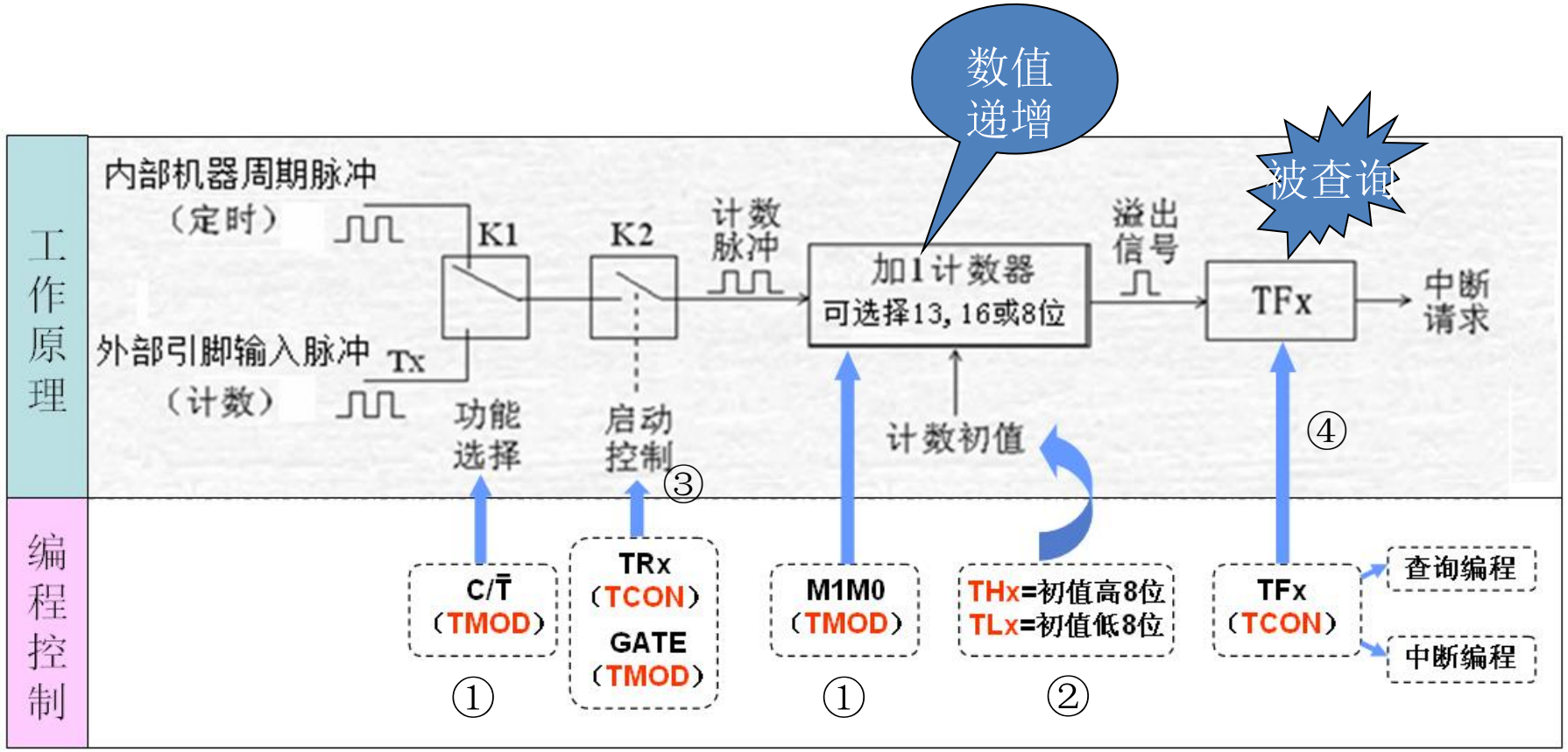
步骤三： 启动定时器

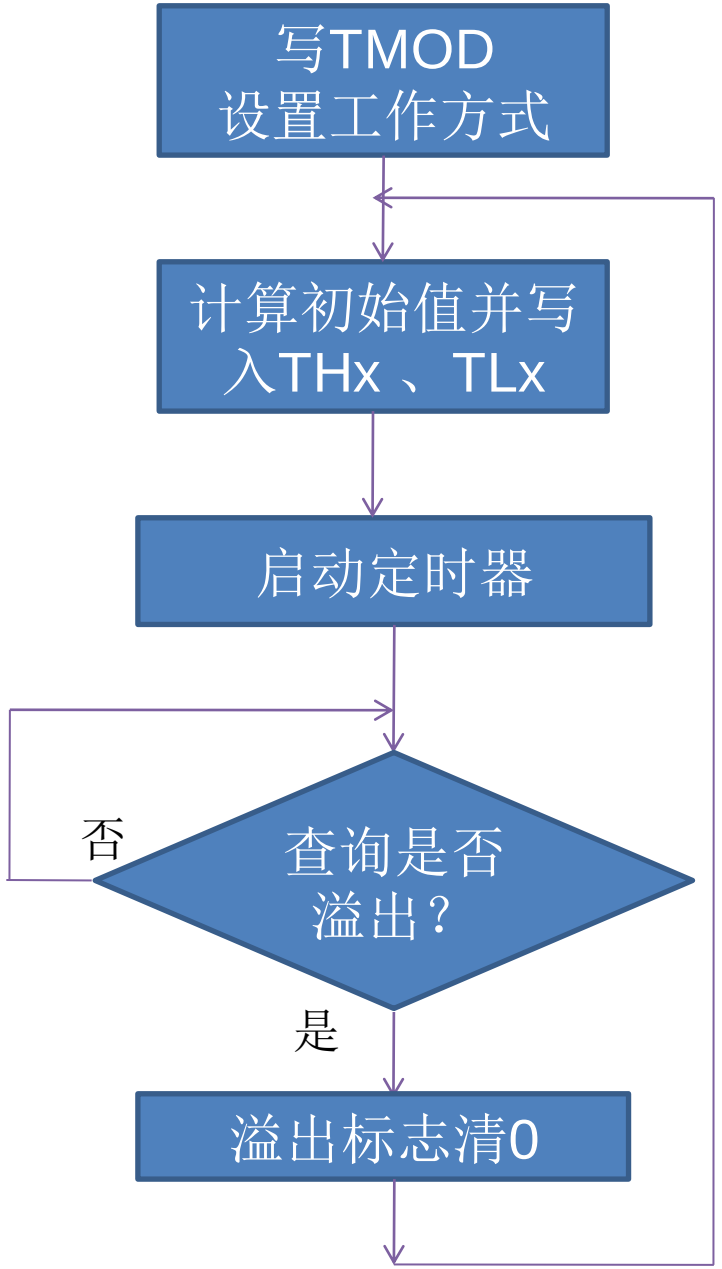




总结——定时器/计数器的编程控制过程

步骤四： 查询是否溢出





总结——定时器/计数器的编程控制过程


```
#include<reg51.h>
void delay1s()  //1s延时函数
{
        unsigned char i;
        for(i=0;i<20;i++)
        {
                TH1=0x3c;
                TL1=0xb0;
                TR1=1;//启动定时器
                //如何处理溢出
                while(!TF1);
                TF1=0;
        }
}
void disp(unsigned char i)//数码管显示函数
{
        unsigned char led[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90} ;//0～9
        P1=led[i/10];   //显示i高位
        P2=led[i%10];   //显示i低位
}
void main()     //主函数
{
   while(1)
    {
                unsigned char miao=0;     //秒计数器定义
                TMOD=0x10;                          //设置T1为工作方式1
                TH1=0x3c;   //设置T1计数初值高8位，定时时间50ms
                TL1=0xb0;   //设置T1计数初值低8位
                TR1=1;                          //启动定时器开始计数
                while(1)
                {
                        disp(miao);                 //显示秒计数器值
                delay1s();                          //调用1s函数
                miao++;                         //秒计数器加1
                        if(miao==100)
                                miao=0;
      }
    }
}
```

提升

1、修改程序，实现任意时间的延时

2、设计时间间隔为1s的流水灯

谢谢聆听!