

浙江大学

《嵌入式系统》 课程实验报告

实验名称：嵌入式系统第二次实验

实验时间：2020.11.9

实验人姓名：胡江鹏

实验人员学号：3180101960

实验人员专业：自动化（控制）

嵌入式系统第二次实验

问题求解:

问题分析

在 Keil 仿真环境中编写程序，实现一个实时时钟；

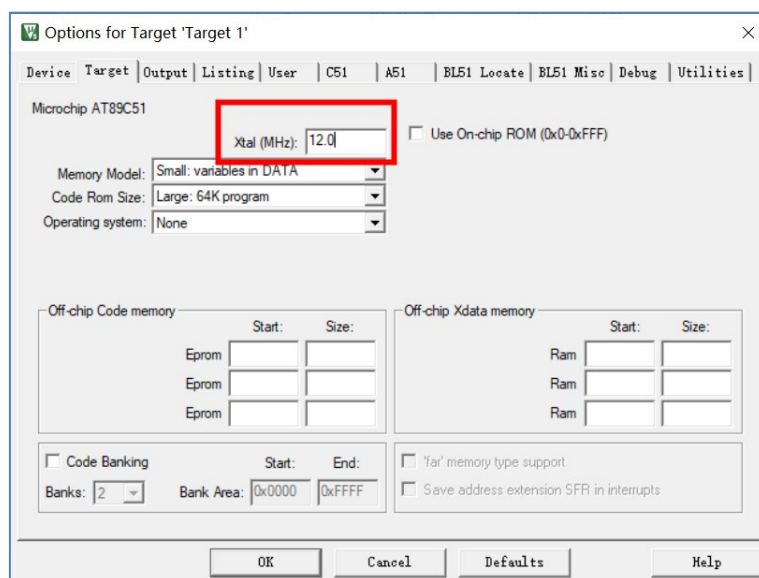
从默认时间点（给定的年月日时分秒）开始计时；必须使用中断响应程序来处理；计时结果（年、月、日、时、分、秒）分别放在内部 RAM 中

实现方法

实现的基本思路为：设定一个基计时器、7 个计数器存储单元（50ms、秒、分、时、日、月、年），在中断函数内进行 7 个计数操作，最后将结果可视化。

1. 基时钟设计

普通时钟的最小计时单位是秒，而单片机的最小计时单位则由其晶振频率决定。本次实验使用的单片机型号为 AT89C51，其内部的晶振频率为 24MHZ，为方便起见，在实践中，通过 keil 仿真软件调节至 12MHZ，如下图所示：



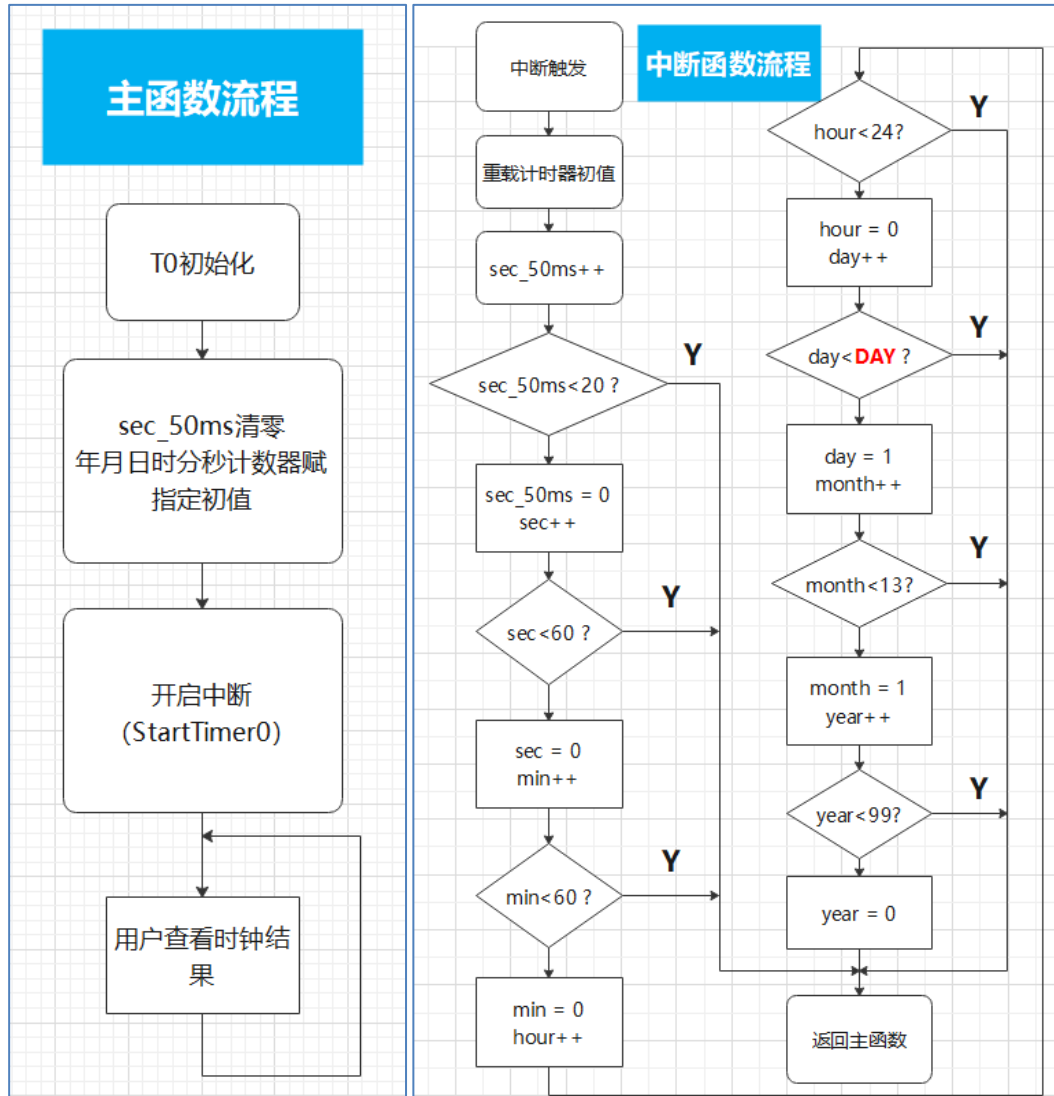
12MHZ 的晶振下最小计时单位为 1us，使用计时器模式 1 时，最长的计时时间约为 65.5ms，为了获得 1s 的计时效果，将计时器的定时时间设置为 50ms，每次中断溢出时通过设定的软件变量 sec_50ms 计数（+1），当 sec_50ms 值为 20 时，即表示定时了 1s。

定时初值：X = 65535 - 50000 = 15535 = 3CB0H

因此定时器赋初值：TH0 = 0X3CH ; TL0 = 0XB0H

2. 软件计数器

在中断函数内部，需要进行软件计数，为了实现年月日时分秒，需要使用分支结构，流程图如下：



其中，计时器重载初值部分使用了增量算法：

TH0 += 0X3C TL0 += 0XB0

如此可以消除中断相应之间微小的时间偏差。

方块图中的 **DAY** 计算涉及不同的月份以及不同的年份（闰年会对 2 月的天数造成影响），因此这里直接使用了查表法，以空间换取时间（牺牲空间以省去 **switch-case** 判断）：事先定义好两行十二列的数组（表），闰年时取行数为 1，非闰年时取行数为 0

```

1. xdata unsigned char tab[2][12] = { //days of month table
2. {31,30,31,28,31,30,31,31,30,31,30,31},
3. {31,30,31,29,31,30,31,31,30,31,30,31} };
  
```

```

1. unsigned char days_of_month(unsigned char month,unsigned char year)
2. {
3.     unsigned int year_all = 2000 + year; // 20xx year
4.     bit flag = 0;
5.     if(year_all % 400 == 0)
6.         flag = 1; //Leap year
7.     else if(year_all % 4 == 0 && year_all % 100 !=0)
8.         flag = 1; //Leap year
9.     return tab[flag][month-1]; //Trade space for time
10.}

```

3. 可视化结果

由于在 keil 中查看寄存器默认为 16 进制显示，为了更好的用户体验，在计算出年月日时分秒后还对其进行了转换计算，使其在调试窗口即可显示十进制的数值。（若要外加数码显示器，也有一定的可迁移性）：

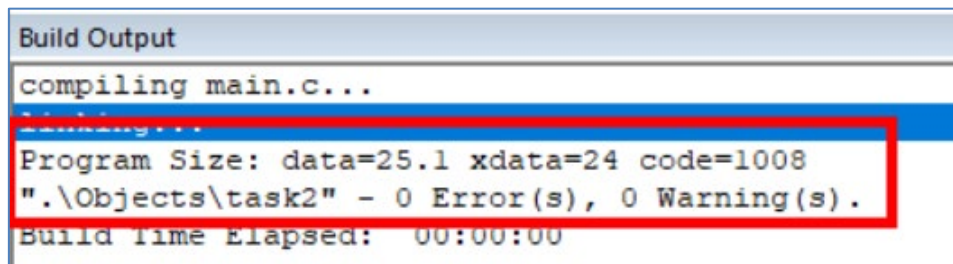
```
1. #define visual(x) (16*((x)/10)+(x)%10)
```

结果展示

调试结果

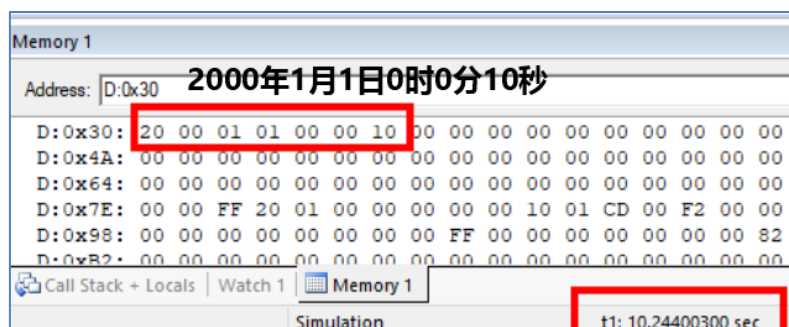
1. 调试验证

指定 2000 年，1 月 1 日，0:00:00，调试结果如下：

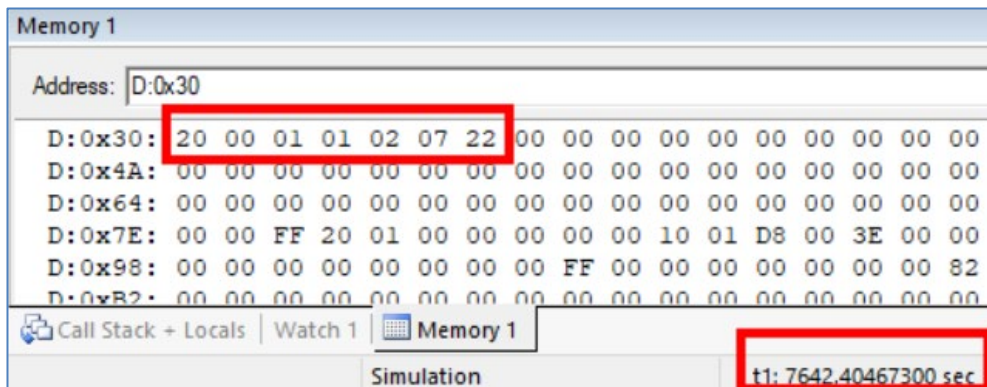


Building Output 中显示了程序的内存大小，ROM:451；内部 RAM：10；外部 RAM：49。

右下角显示了模拟单片机运行时间 sec，通过此数值可以读运行的实际时间，可以与我所实现的实时时钟进行对比。

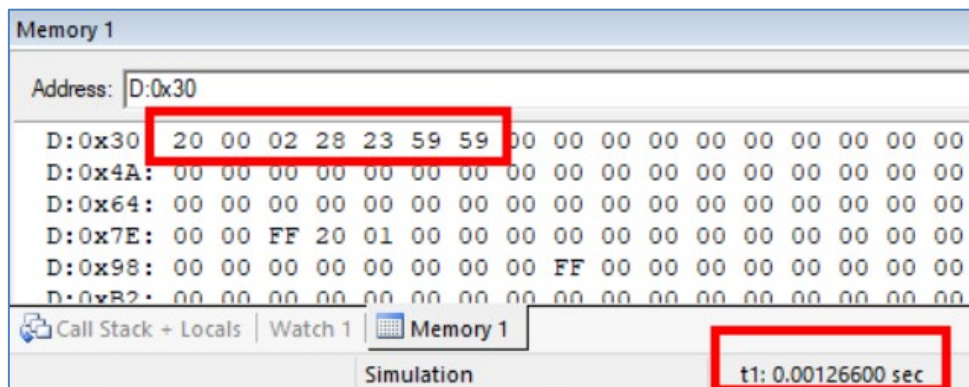


验证时钟、分钟、秒钟可进位:

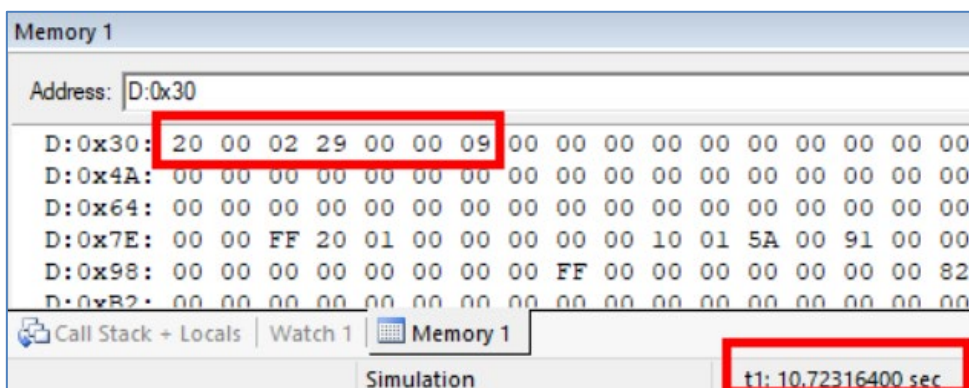


$$7642s = 2h+7min+22s$$

验证闰年: 设置起点 2000 年, 2 月 28 日, 23:59:59



10s 后:



结果分析

整体上看，实时时钟的功能得到实现，且计时较为准确，唯一的缺憾在于占用了单片机较多的内存空间，如果牺牲部分时间效率，可以换取一定的空间，但最终我还是在二者之中选择了牺牲空间，已达到较好的可视效果以及运算效率。

源码附录

```

1. #include<reg51.h>
2. #define RESULT 0x30 //User RAM
3. #define visual(x) (16*((x)/10)+(x)%10) //Decimal number to Hex form-
   visualization
4.
5. code unsigned char* result = RESULT;
6. data unsigned char sec_50ms = 0,sec,min,hour,day,month,year;
7. xdata unsigned char tab[2][12] = { //days of month table
8. {31,30,31,28,31,30,31,31,30,31,30,31},
9. {31,30,31,29,31,30,31,31,30,31,30,31} };
10.
11. // Timer 0 init
12. void StartTimer0(void)
13. {
14. TMOD = 0X01;
15. TH0 = 0X3C;
16. TL0 = 0XB0; // TH0|TL0 = -50000
17. EA = 1;
18. ET0 = 1;
19. TR0 = 1;
20. }
21.
22. unsigned char days_of_month(unsigned char month,unsigned char year)
23. {
24. unsigned int year_all = 2000 + year; // 20xx year
25. bit flag = 0;
26. if(year_all % 400 == 0)
27. flag = 1; //Leap year
28. else if(year_all % 4 == 0 && year_all % 100 !=0)
29. flag = 1; //Leap year
30. return tab[flag][month-1]; //Trade space for time
31. }
32.
33. // Interrupt function
34. void T0Interrupt(void) interrupt 1 using 1
35. {
36. TH0 += 0X3C;
37. TL0 += 0XB0;
38. /*****sec_50ms*****/
39. sec_50ms ++;
40. if(sec_50ms < 20)
41. return;

```

```

42. sec_50ms = 0;
43.
44. /*****sec*****/
45. sec++; result[6] = visual(sec) ;
46. if(sec < 60)
47.     return;
48. sec = result[6] = 0;
49.
50. /*****min*****/
51. min++; result[5] = visual(min) ;
52. if(min < 60)
53.     return;
54. min = result[5] = 0;
55.
56. /*****hour*****/
57. hour++; result[4] = visual(hour);
58. if(hour < 24)
59.     return;
60. hour = result[4] = 0;
61.
62. /*****day*****/
63. day++; result[3] = visual(day);
64. if(day <= days_of_month(month,year))
65.     return;
66. day = result[3] = 1;
67.
68. /*****month*****/
69. month++; result[2] = visual(month);
70. if(month <= 12)
71.     return;
72. month = result[2] = 1;
73.
74. /*****year*****/
75. year++; result[1] = visual(year);
76. if(year < 100) // Out of max recordable time
77.     return;
78. year = result[1] = 0;
79. }
80.
81. void main(void)
82. {
83.     sec = 0; result[6] = visual(sec);
84.     min = 55; result[5] = visual(min);
85.     hour = 23; result[4] = visual(hour);
86.     day = 28; result[3] = visual(day);
87.     month = 2; result[2] = visual(month);
88.     year = 0; result[1] = visual(year);
89.     result[0] = 0x20; // 20xx year
90.     StartTimer0();
91.     while(1);
92. }

```