Embedded System: Assignment #5

Hong Chenhui drredthered.github.io

Zhejiang University — May 1, 2024

Info: Gregorian calendar before January 1st, 1900 is not supported.

1 Problem 1

• Write a perpetual calendar clock using an interrupt service routine, which should start counting from a set time, and store the counting results (year, month, day, hour, minute, second) in an array in RAM.

```
Code as follows:
```

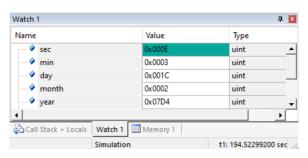
```
#include <reg51.h>
    unsigned int sec_50ms = 0;
    unsigned int sec = 0, min = 59, hour = 23, day = 28, month = 2, year = 2004;
    unsigned int days in month[12] = {31, 28, 31, 30, 31, 30, 31, 30, 31, 30, 31};
    void TimeOlsr(void) interrupt 1
      TH0 = 0x3c;
10
      TL0 = 0xb0;
11
12
      if (sec_50ms == 20)
13
14
        sec 50ms = 0;
15
16
        sec++;
17
        if (sec == 60)
18
19
           sec = 0;
20
21
          min++;
22
           if (min == 60)
23
             min = 0;
25
             hour++;
27
             if (hour == 24)
28
               hour = 0;
30
               day++;
       if ( day == days_in_month[month] + 1 )
33
34
              day = 1;
```

```
month++;
36
        if ( month == 13 )
37
38
                month = 1;
39
                year++;
40
            if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
41
42
         {
           days_in_month[1] = 29;
43
44
         else
45
46
         {
47
           days_in_month[1] = 28;
         }
48
49
50
51
52
         }
53
54
55
         sec_50ms++;
56
57
58
     void main()
59
60
       TMOD = 0x01;
61
       TH0 = 0x3c;
62
       TL0 = 0xb0;
63
       ET0 = 1;

EA = 1;
64
65
       TR0 = 1;
66
67
68
      while (1);
  }
69
```

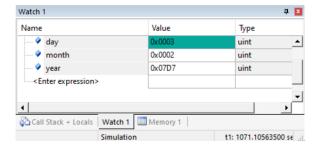
Answer

Running perfectly synchronized with the real time,



 $(194 \sec = 3 \min + 14 \sec = 3:E)$

With $60 \times 60 \times 24$ times (one second per day) result started from 2004-2-28 00:00:00,



after 1071 days, the result is exactly 2007-2-3 00:00:00.