用 51 单片机实现公历与农历、星期的转换

济南　朱思荣

一、概述

公历是全世界通用的历法，以地球绕太阳的一周为一年，一年 365 天，分为 12 个月，1、3、5、7、8、10、12 月为 31 天，2 月为 28 天，其余月份为 30 天。事实上地球绕太阳一周共 365 天 5 小时 48 分 46 秒。比公历一年多出 5 小时 48 分 46 秒，为使年误差不累积，公历年用闰年法来消除年误差， 由于每年多出 5 小时 48 分 46 秒，每 4 年累计多出 23 小时 15 分 4 秒，接近 1 天，天文学家就规定每 4 年有一个闰年，把 2 月由 28 天改为 29 天。凡是公历年代能被 4 整除的那一年就是闰年。但是这样一来，每 4 年又少了 44 分 56 秒，为了更准确地计时，天文学家又规定，凡能被 100 整除的年份，只有能被 400 整除才是闰年，即每 400 年要减掉 3 个闰年，经过这样处理后实际上每 400 年的误差只有 2 小时 53 分 20 秒，已相当准确了。

农历与公历不同，农历把月亮绕地球一周作为一月。因为月亮绕地球一周不是一整天，所以农历把月分为大月和小月，大月 30 天，小月 29 天。通过设置大小月，使农历日始终与月亮与地球的位置相对应。为了使农历的年份与公历年相对应，农历通过设置闰月的办法使它的平均年长度与公历年相等。农历是中国传统文化的代表之一，并与农业生产联系密切，中国人民特别是广大农民十分熟悉并喜爱农历。

公历与农历是我国目前并存的两种历法，各有其固有的规律。农历与月球的运行相对应，其影响因素多，它的大小月和闰月与天体运行有关，计算十分复杂，且每年都不一致。因此要用单片机实现公历与农历的转换，用查表法是最方便实用的办法。

51 系列单片机因其在功能上能满足大部份对速度要求不高的应用场合的要求，且价格低廉，开发工具普及程度高，是目前应用最多的单片机之一。本文介绍一种用 51 单片机实现从 1901 年到 2099 年 199 年公历日到农历日及星期的转换方法，并向读者提供完整的 51 汇编程序。

二、 基本原理

实现公历与农历的转换，一般采用查表法，按日查表是速度最快的方法，但 51 单片机寻址能力有限，不可能采用按日查表的方法。除按日查外，我们可以通过按月查表和按年查表的方法，再通过适当的计算，来确定公历日所对应的农历日期。本文采用的是按年查表法，最大限度地减少表格所占的程序空间。

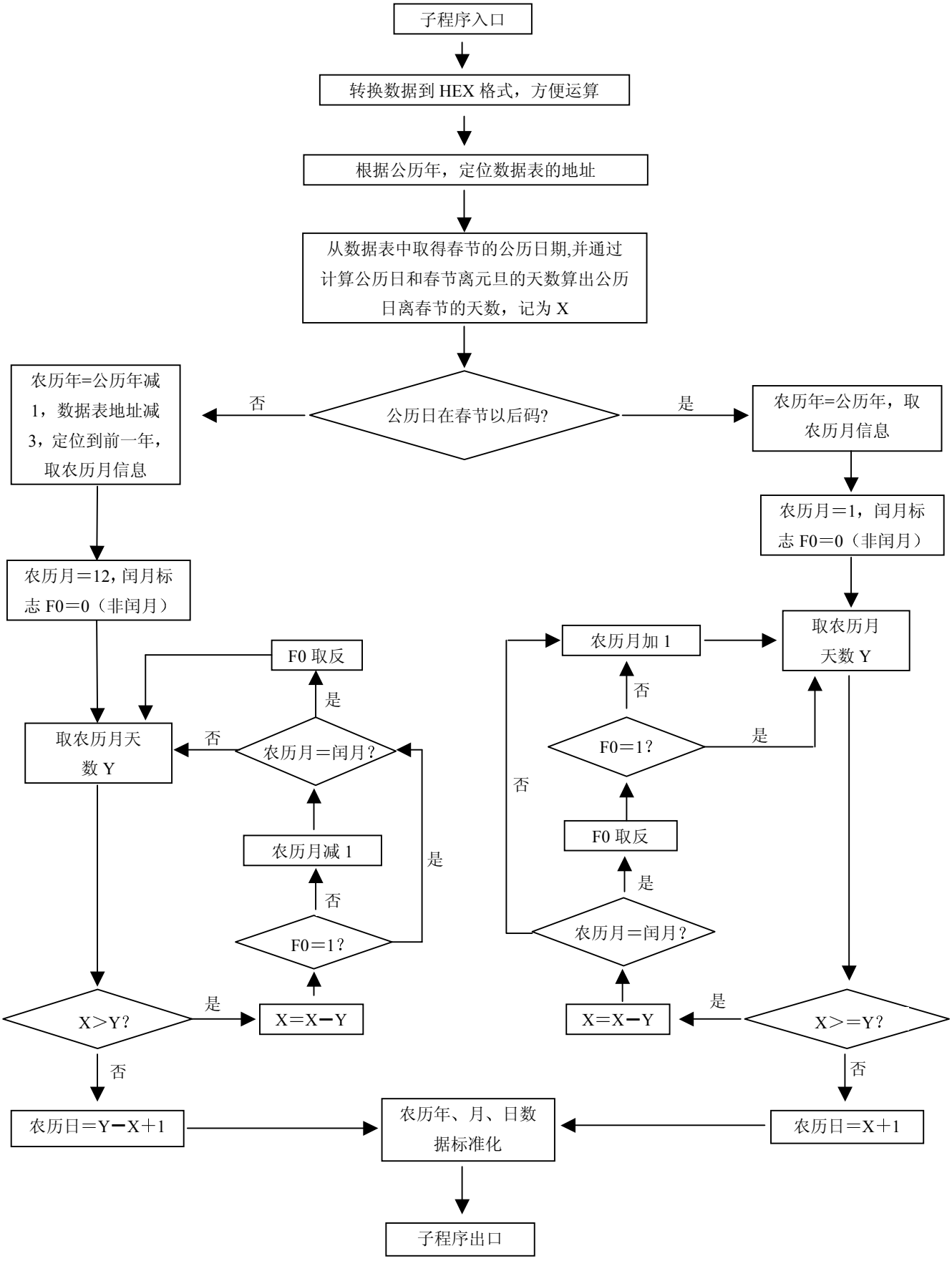
对于农历月来说，大月为 30 天，小月为 29 天，这是固定不变的，这样我们就可用 1 个 BIT（位）来表示大小月信息。农历一年，如有闰月为 13 个月，否则是 12 个月，所以一年需要用 13 个 BIT，闰月在农历年中所在的月份并不固定，大部分闰月分布在农历 2—8 月，但也有少量年份在 9 月以后，所以要表示闰月的信息，至少要 4BIT，在这里我们用 4BIT 的值来表示闰月的月份，值为 0 表示本年没有闰月。有了以上信息，还不足以判断公历日对应的农历日，因为还需要一个参照日，我们选用农历正月初一所对应的公历日期作参照日，公历日最大为 31 日，需要 5BIT 来表示，而春节所在的月份不是 1 月就是 2 月，用 1BIT 就够了，考虑到表达方便，我们用 2BIT 来表示春节月，2BIT 的值直接表示月份。这样一年的农历信息只用 3 个字节就全部包括了。

计算公历日对应的农历日期的方法：先计算出公历日离当年元旦的天数，然后查表取得当年的春节日期，计算出春节离元旦的天数，二者相减即可算出公历日离春节的天数，以后只要根据大小月和闰月信息，减一月天数，调整一月农历月份，即可推算出公历日所对应的农历日期。如公历日不到春节日期，农历年要比公历年小一年，农历大小月取前一年的信息。农历月从 12 月向前推算。

公历日是非常有规律的，所以公历日所对应的星期天可以通过计算直接得到，理论上公元 0 年 1 月 1 日为星期日，只要求得公历日离公元 0 年 1 月 1 日的日子数，除 7 后的余数就是星期天，为了简化计算，采用月校正法，根据公历的年月日可直接计算出星期天。其算法是：日期＋年份＋所过闰年数＋月校正数之和除 7 的余数就是星期天，但如果是在闰年又不到 3 月份，上述之和要减一天再除 7。其 1—12 月的校正数据为：6、2、2、5、0、3、5、1、4、6、2、4。在本程序中采用 1 个字节表示年份，闰年数也只计算 1900 年以后的闰年数，所以实际校正数据也和上述数据不同。

三、 程序流程

由于星期的计算很简单，这里只提供公历日转农历日的程序流程图。



四、程序使用说明

本文提供的子程序在设计时应用了 PCF8563 作时钟芯片，所以其入口格式与 PCF8563 芯片的时钟信号存储格式完全一致，年、月、日均为 BCD 码，其中月的 BIT7 表示世纪，为 1 表示 19 世纪，为 0 表示 20 世纪。采用 PCF8563 时钟芯片只要把它的年、月、日寄存器内容读出到 time_yeAr、time_month 和 time_date 三个单元内即可直接调用本程序转换，采用其它时钟芯片，调用前要把时钟格式稍作调整，或修改一下程序。公历日转农历日程序在 12M 晶振下，执行时间最长约 0.48 毫秒，实际使用时，只需在复位和日期变化时才需要调用一次。对于公历日转星期天的子程序，则只在设置时钟时才有用。在设置时钟年、月、日后调用子程序得到对应的星期天，直接写入时钟即可。

子程序附带了 1901-2100 年的农历数据表，全部编译要占 600 字节空间，如不需这么多可把不需要的数据删除，然后修改 stArt_yeAr 值即可。stArt_yeAr 定义了查询表的起始年份。

五、子程序清单

```
start_year      EQU     01;     定义查询表起始年份,01--199 表示 1901-2099 年
;以下三单元为需转换的公历日期，是子程序的入口数据
time_year       DATA    30h
time_month      DATA    31h     ;BIT7 表示世纪,为 1 表示 19 世纪,为 0 表示 20 世纪
time_date       DATA    32h
;以下三单元存转换后农历日期，与入口单元重叠,如要保留入口信息,请重定义出口地址
CONvert_yeAr    DATA    30h
CONvert_mONth   DATA    31h     ;BIT7 为 1 表示闰月
CONvert_dAte    DATA    32h

temp_Byte1      DATA    37h
temp_Byte2      DATA    38h
temp_Byte3      DATA    39h
temp_Byte4      DATA    3Ah
temp_Byte5      DATA    3Bh

time_week       DATA    40h     ;星期天出口

;以下为公历转农历子程序
CONvert: MOV    A,time_year     ;将年月日转化为 HEX 格式
         MOV    B,#16
         DIV    AB
         MOV    CONvert_yeAr,B
         MOV    B,#10
         MUL    AB
         ADD    A,CONvert_yeAr
         MOV    CONvert_yeAr,A
         MOV    A,time_month
         MOV    C,ACC.7
         MOV    f0,C            ;f0 暂存世纪标志,仅用于数据表定位
         CLR    ACC.7
         JNB    ACC.4,CON_02
```

```
        CLR     ACC.4           ;ACC.4 为 1 表示大于 10 月
        ADD     A,#10
CON_02: MOV     CONvert_mONth,A
        MOV     A,time_date
        MOV     B,#16
        DIV     AB
        MOV     CONvert_dAte,B
        MOV     B,#10
        MUL     AB
        ADD     A,CONvert_dAte
        MOV     CONvert_dAte,A

        MOV     dptr,#mONth_dAtA ;以下定位本年数据在表格中的位置
        MOV     A,CONvert_yeAr
        JB      f0,CON_06       ;当前为 19 世纪年跳转
        ADD     A,#100          ;从 19 世纪起定义表格起始年,20 世纪要加 100 年
CON_06: CLR     C
        SUBB    A,#stArt_yeAr
        MOV     B,#3            ;表格每年 3 字节
        MUL     AB
        ADD     A,dpl
        MOV     dpl,A
        MOV     A,B
        ADDC    A,dph
        MOV     dph,A
        MOV     A,#2
        MOVC    A,@A+dptr       ;读本年表格最后一字节(春节日期)
        CLR     ACC.7           ;ACC.7 是闰年第 13 个月大小,在此不用
        MOV     B,#32
        DIV     AB
        MOV     temp_Byte1,A    ;春节月份
        MOV     temp_Byte2,B    ;春节日

        ;以下计算当前日期距元旦天数
        MOV     temp_Byte3,#0   ;设距元旦天数高位为 0
        MOV     A,CONvert_mONth
        CJNE    A,#10,CON_08
CON_08: JC      CON_09          ;9 月以前日子数小于 256 天,高字节为 0(9 月份过去的整月为 8 个月)
        MOV     temp_Byte3,#1
CON_09: MOV     A,CONvert_yeAr
        ANL     A,#03h          ;ACC 为除 4 的余数
        JNZ     CON_10          ;转常年处理
        ;年除 4 余数为 0 是闰年
        MOV     A,CONvert_mONth
```

```
        LCALL   get_ruN_dAys_lOw        ;取得闰年过去月的天数的低字节
        SJMP    CON_12
CON_10: MOV     A,CONvert_mONth
        LCALL   get_dAys_lOw            ;取得常年过去月的天数的低字节
CON_12: MOV     B,CONvert_dAte
        DEC     B                       ;因为日期从 1 日起,而不是 0 日起
        ADD     A,B                     ;过去的整月天数加当月天数
        MOV     temp_Byte4,A
        JNC     CON_14
        INC     temp_Byte3              ;temp_Byte3,temp_Byte4 分别为公历年过去的天数的高低字节

        ;以下求春节距元旦天数,因肯定小于 256 天所以只用一字节表示
CON_14: MOV     A,temp_Byte1
        LCALL   get_dAys_lOw            ;春节不会在 3 月份,不用考虑闰年
        DEC     A                       ;因为日期从 1 日起
        ADD     A,temp_Byte2
        MOV     temp_Byte5,A            ;temp_Byte5,为春节距元旦天数

        MOV     A,CONvert_mONth
        CJNE    A,temp_Byte1,CON_20     ;转换月与春节月比较
        MOV     A,CONvert_dAte
        CJNE    A,temp_Byte2,CON_20     ;转换日与春节日比较
CON_20: JC      CON_22
        LJMP    CON_60                  ;当前日大于等于春节日期,公历年与农历年同年份

CON_22: MOV     A,CONvert_yeAr          ;不到春节,农历年比公历年低一年
        JNZ     CON_24
        MOV     A,#100                  ;年有效数 0-99
CON_24: DEC     A
        MOV     CONvert_yeAr,A
        MOV     A,dpl
        CLR     C
        SUBB    A,#3
        MOV     dpl,A
        JNC     CON_26
        DEC     dph                     ;表格指针指向上一年
CON_26: MOV     A,temp_Byte5
        CLR     C
        SUBB    A,temp_Byte4
        MOV     temp_Byte3,A            ;temp_Byte3 中为当前日离春节的天数
        MOV     CONvert_mONth,#12       ;农历月为 12 月
        CLR     f0                      ;1901-2099 年没有闰 12 月,清闰月标志
        CLR     A
        MOVC    A,@A+dptr
```

```
        ANL     A,#0f0h
        SWAP    A;
        MOV     temp_Byte4,A        ;temp_Byte4 中为闰月
        JZ      CON_30              ;没有闰月转移
        MOV     A,#2                ;有闰月,取第 13 个月天数
        MOVC    A,@A+dptr
        MOV     C,ACC.7
        MOV     A,#1
        MOVC    A,@A+dptr
        RLC     A                   ;ACC 中为最后 6 个月的大小值
        SJMP    CON_34
CON_30: MOV     A,#1
        MOVC    A,@A+dptr           ;ACC 中为最后 6 个月的大小值
CON_34: MOV     temp_Byte5,A

CON_40: MOV     A,temp_Byte5
        RRC     A
        MOV     temp_Byte5,A
        JC      CON_42
        MOV     B,#29               ;小月 29 天
        SJMP    CON_44
CON_42: MOV     B,#30               ;大月 30 天
CON_44: MOV     A,temp_Byte3
        CLR     C
        SUBB    A,B
        JZ      CON_46              ;正好够减,就是农历日 1 日
        JNC     CON_50
        ;不够减一月天数,结束农历月调整
        CPL     A                   ;求补取绝对值
        INC     A
CON_46: INC     A                   ;加 1 即为农历日
        MOV     B,#10               ;转换并保存农历日,月,年
        DIV     AB
        SWAP    A
        ORL     A,B
        MOV     CONvert_dAte,A
        MOV     A,CONvert_mONth
        MOV     B,#10
        DIV     AB
        SWAP    A
        ORL     A,B
        MOV     C,f0
        MOV     ACC.7,C
        MOV     CONvert_mONth,A
```

```
        MOV     A,CONvert_yeAr
        MOV     B,#10
        DIV     AB
        SWAP    A
        ORL     A,B
        MOV     CONvert_yeAr,A
        RET                                 ;结束转换

CON_50: MOV     temp_Byte3,A                ;temp_Byte3 存减去一月后的天数
        JB      f0,CON_52                   ;是闰月,前推一月,月份不减
        DEC     CONvert_mONth;
CON_52: MOV     A,CONvert_mONth
        CJNE    A,temp_Byte4,CON_54
        CPL     f0                          ;当前月与闰月相同,更改闰月标志
CON_54: SJMP    CON_40

CON_60: MOV     A,temp_Byte4                ;春节日小于当前日,农历年同公历年
        CLR     C
        SUBB    A,temp_Byte5
        MOV     temp_Byte4,A
        JNC     CON_62
        DEC     temp_Byte3                  ;temp_Byte3 temp_Byte4 中为公历日离春节的天数
CON_62: MOV     CONvert_mONth,#1            ;农历月为 1 月
        CLR     A
        MOVC    A,@A+dptr
        MOV     temp_Byte5,A
        ANL      A,#0f0h
        SWAP    A;
        XCH     A,temp_Byte5                ;temp_Byte5 中为闰月,ACC 为当年农历表第一字节
        CLR     f0                          ;第一个月肯定不是闰月
        ANL     A,#0fh
        MOV     temp_Byte1,A
        MOV     A,#1
        MOVC    A,@A+dptr
        MOV     temp_Byte2,A
        ANL     A,#0f0h
        ORL     A,temp_Byte1
        SWAP    A
        MOV     temp_Byte1,A
        MOV     A,#2
        MOVC    A,@A+dptr
        MOV     C,ACC.7
        MOV     A,temp_Byte2
        ANL     A,#0fh
```

```
        SWAP    A
        MOV     ACC.3,C;
        MOV     temp_Byte2,A                ;以上 temp_Byte1,temp_Byte2 各 BIT 存农历年大小
CON_70: MOV     A,temp_Byte2
        RLC     A
        MOV     temp_Byte2,A
        MOV     A,temp_Byte1
        RLC     A
        MOV     temp_Byte1,A
        JC      CON_72
        MOV     B,#29                       ;小月 29 天处理
        SJMP    CON_74
CON_72: MOV     B,#30                       ;大月 30 天
CON_74: MOV     A,temp_Byte4
        CLR     C
        SUBB    A,B
        JNC     CON_78                      ;低字节够减跳转
        MOV     B,A                         ;低字节不够减, B 暂存减后结果,
        MOV     A,temp_Byte3
        JZ      CON_76                      ;高字节为 0,不够减
        DEC     temp_Byte3
        MOV     temp_Byte4,B
        SJMP    CON_80
CON_76: MOV     A,temp_Byte4                ;不够减结束月调整
        LJMP    CON_46                      ;转日期加 1 后,处理并保存转换后农历年月日
CON_78: MOV     temp_Byte4,A                ;temp_Byte3 temp_Byte4 天数为减去一月后天数
CON_80: MOV     A,CONvert_mONth
        CJNE    A,temp_Byte5,CON_82
        CPL     f0                          ;当前月与闰月相同,更改闰月标志
        JNB     f0,CON_82                   ;更改标志后是非闰月,月份加 1
        SJMP    CON_70
CON_82: INC     CONvert_mONth;
        SJMP    CON_70
get_dAys_lOw:
        MOVC    A,@A+PC                     ;取得常年过去月的天数的低字节
        RET
        DB 0,31,59,90,120,151,181,212,243,17,48,78
get_ruN_dAys_lOw:
        MOVC    A,@A+PC                     ;取得闰年过去月的天数的低字节
        RET
        DB 0,31,60,91,121,152,182,213,244,18,49,79

mONth_dAtA:
;公历年对应的农历数据,每年三字节,
```

;格式第一字节 BIT7-4 位表示闰月月份,值为 0 为无闰月,BIT3-0 对应农历第 1-4 月的大小
;第二字节 BIT7-0 对应农历第 5-12 月大小,第三字节 BIT7 表示农历第 13 个月大小
;月份对应的位为 1 表示本农历月大(30 天),为 0 表示小(29 天).
;第三字节 BIT6-5 表示春节的公历月份,BIT4-0 表示春节的公历日期

```
DB 004h,0Aeh,053h;    1901;
DB 00Ah,057h,048h;    1902
DB 055h,026h,0Bdh;    1903
DB 00dh,026h,050h;    1904
DB 00dh,095h,044h;    1905
DB 046h,0AAh,0B9h;    1906
DB 005h,06Ah,04dh;    1907
DB 009h,0Adh,042h;    1908
DB 024h,0Aeh,0B6h;    1909
DB 004h,0Aeh,04Ah;    1910
DB 06Ah,04dh,0Beh;    1911
DB 00Ah,04dh,052h;    1912
DB 00dh,025h,046h;    1913
DB 05dh,052h,0BAh;    1914
DB 00Bh,054h,04eh;    1915
DB 00dh,06Ah,043h;    1916
DB 029h,06dh,037h;    1917
DB 009h,05Bh,04Bh;    1918
DB 074h,09Bh,0C1h;    1919
DB 004h,097h,054h;    1920
DB 00Ah,04Bh,048h;    1921
DB 05Bh,025h,0BCh;    1922
DB 006h,0A5h,050h;    1923
DB 006h,0d4h,045h;    1924
DB 04Ah,0dAh,0B8h;    1925
DB 002h,0B6h,04dh;    1926
DB 009h,057h,042h;    1927
DB 024h,097h,0B7h;    1928
DB 004h,097h,04Ah;    1929
DB 066h,04Bh,03eh;    1930
DB 00dh,04Ah,051h;    1931
DB 00eh,0A5h,046h;    1932
DB 056h,0d4h,0BAh;    1933
DB 005h,0Adh,04eh;    1934
DB 002h,0B6h,044h;    1935
DB 039h,037h,038h;    1936
DB 009h,02eh,04Bh;    1937
DB 07Ch,096h,0Bfh;    1938
DB 00Ch,095h,053h;    1939
DB 00dh,04Ah,048h;    1940
```

```
DB 06dh,0A5h,03Bh;  1941
DB 00Bh,055h,04fh;  1942
DB 005h,06Ah,045h;  1943
DB 04Ah,0Adh,0B9h;  1944
DB 002h,05dh,04dh;  1945
DB 009h,02dh,042h;  1946
DB 02Ch,095h,0B6h;  1947
DB 00Ah,095h,04Ah;  1948
DB 07Bh,04Ah,0Bdh;  1949
DB 006h,0CAh,051h;  1950
DB 00Bh,055h,046h;  1951
DB 055h,05Ah,0BBh;  1952
DB 004h,0dAh,04eh;  1953
DB 00Ah,05Bh,043h;  1954
DB 035h,02Bh,0B8h;  1955
DB 005h,02Bh,04Ch;  1956
DB 08Ah,095h,03fh;  1957
DB 00eh,095h,052h;  1958
DB 006h,0AAh,048h;  1959
DB 07Ah,0d5h,03Ch;  1960
DB 00Ah,0B5h,04fh;  1961
DB 004h,0B6h,045h;  1962
DB 04Ah,057h,039h;  1963
DB 00Ah,057h,04dh;  1964
DB 005h,026h,042h;  1965
DB 03eh,093h,035h;  1966
DB 00dh,095h,049h;  1967
DB 075h,0AAh,0Beh;  1968
DB 005h,06Ah,051h;  1969
DB 009h,06dh,046h;  1970
DB 054h,0Aeh,0BBh;  1971
DB 004h,0Adh,04fh;  1972
DB 00Ah,04dh,043h;  1973
DB 04dh,026h,0B7h;  1974
DB 00dh,025h,04Bh;  1975
DB 08dh,052h,0Bfh;  1976
DB 00Bh,054h,052h;  1977
DB 00Bh,06Ah,047h;  1978
DB 069h,06dh,03Ch;  1979
DB 009h,05Bh,050h;  1980
DB 004h,09Bh,045h;  1981
DB 04Ah,04Bh,0B9h;  1982
DB 00Ah,04Bh,04dh;  1983
DB 0ABh,025h,0C2h;  1984
```

```
DB 006h,0A5h,054h;  1985
DB 006h,0d4h,049h;  1986
DB 06Ah,0dAh,03dh;  1987
DB 00Ah,0B6h,051h;  1988
DB 009h,037h,046h;  1989
DB 054h,097h,0BBh;  1990
DB 004h,097h,04fh;  1991
DB 006h,04Bh,044h;  1992
DB 036h,0A5h,037h;  1993
DB 00eh,0A5h,04Ah;  1994
DB 086h,0B2h,0Bfh;  1995
DB 005h,0ACh,053h;  1996
DB 00Ah,0B6h,047h;  1997
DB 059h,036h,0BCh;  1998
DB 009h,02eh,050h;  1999
DB 00Ch,096h,045h;  2000
DB 04dh,04Ah,0B8h;  2001
DB 00dh,04Ah,04Ch;  2002
DB 00dh,0A5h,041h;  2003
DB 025h,0AAh,0B6h;  2004
DB 005h,06Ah,049h;  2005
DB 07Ah,0Adh,0Bdh;  2006
DB 002h,05dh,052h;  2007
DB 009h,02dh,047h;  2008
DB 05Ch,095h,0BAh;  2009
DB 00Ah,095h,04eh;  2010
DB 00Bh,04Ah,043h;  2011
DB 04Bh,055h,037h;  2012
DB 00Ah,0d5h,04Ah;  2013
DB 095h,05Ah,0Bfh;  2014
DB 004h,0BAh,053h;  2015
DB 00Ah,05Bh,048h;  2016
DB 065h,02Bh,0BCh;  2017
DB 005h,02Bh,050h;  2018
DB 00Ah,093h,045h;  2019
DB 047h,04Ah,0B9h;  2020
DB 006h,0AAh,04Ch;  2021
DB 00Ah,0d5h,041h;  2022
DB 024h,0dAh,0B6h;  2023
DB 004h,0B6h,04Ah;  2024
DB 069h,057h,03dh;  2025
DB 00Ah,04eh,051h;  2026
DB 00dh,026h,046h;  2027
DB 05eh,093h,03Ah;  2028
```

```
DB 00dh,053h,04dh;    2029
DB 005h,0AAh,043h;    2030
DB 036h,0B5h,037h;    2031
DB 009h,06dh,04Bh;    2032
DB 0B4h,0Aeh,0Bfh;    2033
DB 004h,0Adh,053h;    2034
DB 00Ah,04dh,048h;    2035
DB 06dh,025h,0BCh;    2036
DB 00dh,025h,04fh;    2037
DB 00dh,052h,044h;    2038
DB 05dh,0AAh,038h;    2039
DB 00Bh,05Ah,04Ch;    2040
DB 005h,06dh,041h;    2041
DB 024h,0Adh,0B6h;    2042
DB 004h,09Bh,04Ah;    2043
DB 07Ah,04Bh,0Beh;    2044
DB 00Ah,04Bh,051h;    2045
DB 00Ah,0A5h,046h;    2046
DB 05Bh,052h,0BAh;    2047
DB 006h,0d2h,04eh;    2048
DB 00Ah,0dAh,042h;    2049
DB 035h,05Bh,037h;    2050
DB 009h,037h,04Bh;    2051
DB 084h,097h,0C1h;    2052
DB 004h,097h,053h;    2053
DB 006h,04Bh,048h;    2054
DB 066h,0A5h,03Ch;    2055
DB 00eh,0A5h,04fh;    2056
DB 006h,0B2h,044h;    2057
DB 04Ah,0B6h,038h;    2058
DB 00Ah,0Aeh,04Ch;    2059
DB 009h,02eh,042h;    2060
DB 03Ch,097h,035h;    2061
DB 00Ch,096h,049h;    2062
DB 07dh,04Ah,0Bdh;    2063
DB 00dh,04Ah,051h;    2064
DB 00dh,0A5h,045h;    2065
DB 055h,0AAh,0BAh;    2066
DB 005h,06Ah,04eh;    2067
DB 00Ah,06dh,043h;    2068
DB 045h,02eh,0B7h;    2069
DB 005h,02dh,04Bh;    2070
DB 08Ah,095h,0Bfh;    2071
DB 00Ah,095h,053h;    2072
```

```
        DB 00Bh,04Ah,047h;   2073
        DB 06Bh,055h,03Bh;   2074
        DB 00Ah,0d5h,04fh;   2075
        DB 005h,05Ah,045h;   2076
        DB 04Ah,05dh,038h;   2077
        DB 00Ah,05Bh,04Ch;   2078
        DB 005h,02Bh,042h;   2079
        DB 03Ah,093h,0B6h;   2080
        DB 006h,093h,049h;   2081
        DB 077h,029h,0Bdh;   2082
        DB 006h,0AAh,051h;   2083
        DB 00Ah,0d5h,046h;   2084
        DB 054h,0dAh,0BAh;   2085
        DB 004h,0B6h,04eh;   2086
        DB 00Ah,057h,043h;   2087
        DB 045h,027h,038h;   2088
        DB 00dh,026h,04Ah;   2089
        DB 08eh,093h,03eh;   2090
        DB 00dh,052h,052h;   2091
        DB 00dh,0AAh,047h;   2092
        DB 066h,0B5h,03Bh;   2093
        DB 005h,06dh,04fh;   2094
        DB 004h,0Aeh,045h;   2095
        DB 04Ah,04eh,0B9h;   2096
        DB 00Ah,04dh,04Ch;   2097
        DB 00dh,015h,041h;   2098
        DB 02dh,092h,0B5h;   2099
        DB 00dh,053h,049h;   2100

;以下子程序用于从当前公历日期,推算星期,
;入口:time_yeAr,time_month      ,time_date  ,定义公历年、月、日,BCD 码,其中月的
;BIT7 表示世纪,0 表示 20 世纪,1 表示 19 世纪,与 PCF8563 一致。
;出口  time_week, 0-6 表示星期日-星期六,与 PCF8563 一致,程序不改变入口数据。
;使用资源:ACC,B,psw,temp_Byte1,temp_Byte2,temp_Byte3。
GetWeek: MOV     A,time_yeAr
        MOV     B,#16
        DIV     AB
        MOV     temp_Byte1,B
        MOV     B,#10
        MUL     AB
        ADD     A,temp_Byte1
        MOV     temp_Byte1,A                    ;temp_Byte1=年
        MOV     A,time_month
        JB      ACC.7,getw02
```

```
        MOV     A,#100
        ADD     A,temp_Byte1
        MOV     temp_Byte1,A            ;20 世纪年+100
        MOV     A,time_month
        CLR     ACC.7
getw02: JNB     ACC.4,getw04
        ADD     A,#10
        CLR     ACC.4
getw04: MOV     temp_Byte2,A            ;temp_Byte2=月
        MOV     A,time_date
        MOV     B,#16
        DIV     AB
        MOV     temp_Byte3,B
        MOV     B,#10
        MUL     AB
        ADD     A,temp_Byte3
        MOV     temp_Byte3,A            ;temp_Byte3=日
         MOV    A,temp_Byte1;
        ANL     A,#03h
        JNZ     getw10                  ;非闰年转移
        MOV     A,temp_Byte2
        CJNE    A,#3,getw06
getw06: JNC     getw10                  ;月大于 2 转移
        DEC     temp_Byte3              ;份小于等于 2,又是闰年,日减 1
getw10: MOV     A,temp_Byte2;
        LCALL   get_CorreCt             ;取月校正表数据
        ADD     A,temp_Byte1
        MOV     B,#7
        DIV     AB                      ;B 放年加校正日数之和后除 7 的余数，不先做这一步，
                                        ;有可能数据溢出。
        MOV     A,temp_Byte1
        ANL     A,#0fCh
        RR      A
        RR      A                       ;以上年除 4 即闰年数
        ADD     A,B
        ADD     A,temp_Byte3
        MOV     B,#7
        DIV     AB
        MOV     time_week,B
        RET
get_COrreCt:
        MOVC    A,@A+PC
        RET
        DB 0,3,3,6,1,4,6,2,5,0,3,5
```