

## 带 32 位 MCU 和高精度 ADC 的 SoC 产品

### ----SD93F 系列开发指南（五）

#### 一、前言

在开发指南（四）中，我们简单介绍了 PWM 功能的使用，本章我们再结合 LCD 显示功能介绍一下实时时钟 RTC 功能，参考应用手册第 14 章节。

#### 二、RTC 结构图：

我们首先来看一下 RTC 的功能框图，如下图 1：

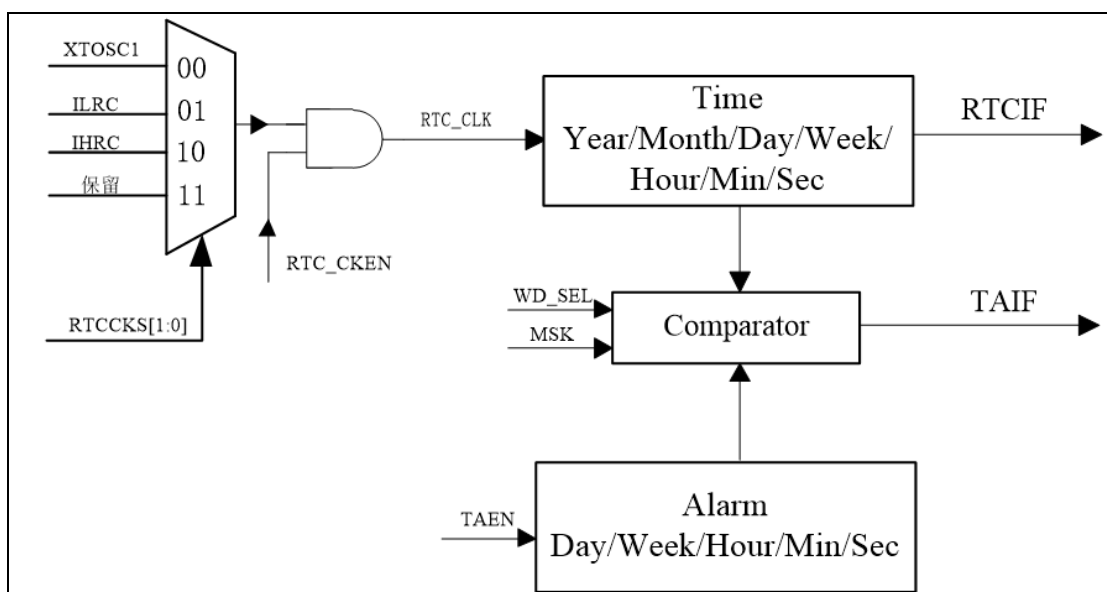


图 1.RTC 结构图

从图中我们看到 RTC 模块整体分为 RTC 时钟源选择、时间初值设置、闹钟设置、中断控制和 RTC 使能控制构成，在时钟源、时间初值、闹钟和中断都配置完成后打开 RTC 使能，即可开始自动计时。

#### 三、RTC 计时显示：

接下来按照 RTC 初始化函数（见图 2）简单介绍一下 RTC 的配置过程，第一步先打开 RTC 外设时钟使能信号，接下来选择 RTC 的时钟源，时钟源的精度会非常影响 RTC 计时的准确度，因此一般都选用外部 32.768kHz 晶体作为 RTC 的时钟源，为了演示方便我们选择用内部 IHRC 作为时钟源，然后设置时间初值，为了直观的看到年、月、日、星期、时、分、秒的每个数据都能发生变化，示例以 24 小时制设定时间初值为 20 年 12 月 31 日，23 时 59 分 40 秒，使能 RTC 半秒中断（中断设置过程可以参考开发指南（二）中按键中断的相关内容），最后再使能 RTC。

```

main.c  RTC.c X SD931X_it.c  LCD.c  Startup_SD931XS  delay.c  SD931X.h  define.h  SD931X_extic  SD931X_rtc.h
1      #include "define.h"
2
3      void RTC_MyInit(void)
4      {
5          RTC_InitTypeDef RTC_InitStruct;
6          VIC_InitTypeDef VIC_InitType;
7
8          RCC_APBPeriphClockCmd(RCC_APBPeriph_RTC, ENABLE); //外设RTC时钟使能
9          RCC_APB2PeriphClockConfig(RCC_APBPeriph_RTC, RCC_RTCCLK_IHRC, RCC_PVMCLK_DIV1); //选择RTC的时钟源为IHRC
10
11         RTC_InitStruct.RTC_HourFormat=RTC_HourFormat_24; //选择24小时制
12         RTC_InitStruct.RTC_DateStruct.RTC_Year=0x20; //设置日期,年:20
13         RTC_InitStruct.RTC_DateStruct.RTC_Month=RTC_Month_December; //设置日期,月:12
14         RTC_InitStruct.RTC_DateStruct.RTC_Date=0x31; //设置日期,日:31
15         RTC_InitStruct.RTC_TimeStruct.RTC_Hours=0x23; //设置时间,小时:23
16         RTC_InitStruct.RTC_TimeStruct.RTC_Minutes=0x59; //设置时间,分钟:59
17         RTC_InitStruct.RTC_TimeStruct.RTC_Seconds=0x40; //设置时间,秒:40
18         RTC_StructInit(&RTC_InitStruct);
19
20         //中断设置
21         VIC_InitType.VIC_IRQChannel=RTC_IRQn;
22         VIC_InitType.VIC_IRQChannelCmd=ENABLE; //使能对应编号的中断通道
23         VIC_InitType.VIC_IRQChannelPriority=VIC_IRQChannelPriority1; //优先级设置为1
24         VIC_InitType.VIC_IRQChannelWakeUpLowPowerCmd=DISABLE; //不使能中断唤醒低功耗
25         VIC_InitType.VIC_PrioThreshold=VIC_PrioThreshold1; //阈值设置为1
26         VIC_InitType.VIC_PrioThresholdCmd=ENABLE; //使能优先级阈值比较功能
27         VIC_Init(&VIC_InitType); //中断初始化
28
29         RTC_ClearITStatus(RTC_IT_RTCIE); //清零半秒中断标志位
30         RTC_ITConfig(RTC_IT_RTCIE, ENABLE); //RTC半秒中断使能
31
32         RTC_Cmd(ENABLE); //RTC使能
33         count=0;
34     }

```

图 2.RTC 初始化函数

RTC 初始化完成后我们再验证一下 RTC 的计数功能（如图 3），这里我们把显示时间的部分程序放在 RTC 半秒中断里执行，因此每 0.5S 就会进入一次中断，进入中断后要先将中断标志位清除，然后判断 RTC\_FLAG 标志位（按键中断程序里设定的 Key2 按下时置起的标志位），当 RTC\_FLAG 标志位置 1 后，读取日期和时间数据，再调用 LCD 显示函数，就能在 LCD 屏上显示正在计时的时间了。由于 LCD 屏的显示内容有限，这里我们先显示 5S 的时分秒（如图 4），再显示半秒的月日星期（如图 5），就能看到较完整的 RTC 计时显示了。

```

FLAG.c  SD931X_jwdg.c  Startup_SD931XS  Sys.c  LCD.c  delay.c  SD931X_lcd.h  SD931X_pwr.c
273     void RTC_IRQHandler(void)
274     {
275         RTC_DateTypeDef RTC_DateStruct;
276         RTC_TimeTypeDef RTC_TimeStruct;
277         RTC_ClearITStatus(RTC_IT_RTCIE); //清半秒中断标志位
278         if(RTC_FLAG)
279         {
280             count++;
281             if(count <10) //显示5秒时间
282             {
283                 RTC_GetTime(&RTC_TimeStruct); //获取时间
284                 LCD_ClearDisAll(); //清显存
285                 LCD_DisplayData(1, RTC_TimeStruct.RTC_Hours>>4 &0X0F); //显示小时
286                 LCD_DisplayData(2, RTC_TimeStruct.RTC_Hours &0X0F); //显示小时
287                 LCD_DisplayData(3, RTC_TimeStruct.RTC_Minutes>>4 &0X0F); //显示分钟
288                 LCD_DisplayData(4, RTC_TimeStruct.RTC_Minutes &0X0F); //显示分钟
289                 LCD_DisplayData(5, RTC_TimeStruct.RTC_Seconds>>4 &0X0F); //显示秒
290                 LCD_DisplayData(6, RTC_TimeStruct.RTC_Seconds &0X0F); //显示秒
291             }
292             else if(count == 11) //显示半秒日期
293             {
294                 RTC_GetDate(&RTC_DateStruct); //获取日期
295                 LCD_ClearDisAll(); //清显存
296                 LCD_DisplayData(1, RTC_DateStruct.RTC_Month>>4 &0X0F); //显示月
297                 LCD_DisplayData(2, RTC_DateStruct.RTC_Month &0X0F); //显示月
298                 LCD_DisplayData(3, RTC_DateStruct.RTC_Date>>4 &0X0F); //显示日
299                 LCD_DisplayData(4, RTC_DateStruct.RTC_Date &0X0F); //显示日
300                 if(RTC_DateStruct.RTC_WeekDay != 0x00)
301                 {
302                     LCD_DisplayData(6, RTC_DateStruct.RTC_WeekDay &0X0F); //显示星期一~星期六
303                 }
304                 else
305                 {
306                     LCD_DisplayData(6, 0X07); //显示星期天
307                 }
308                 count =0;
309             }
310         }
311     }

```

图 3.RTC 中断服务函数

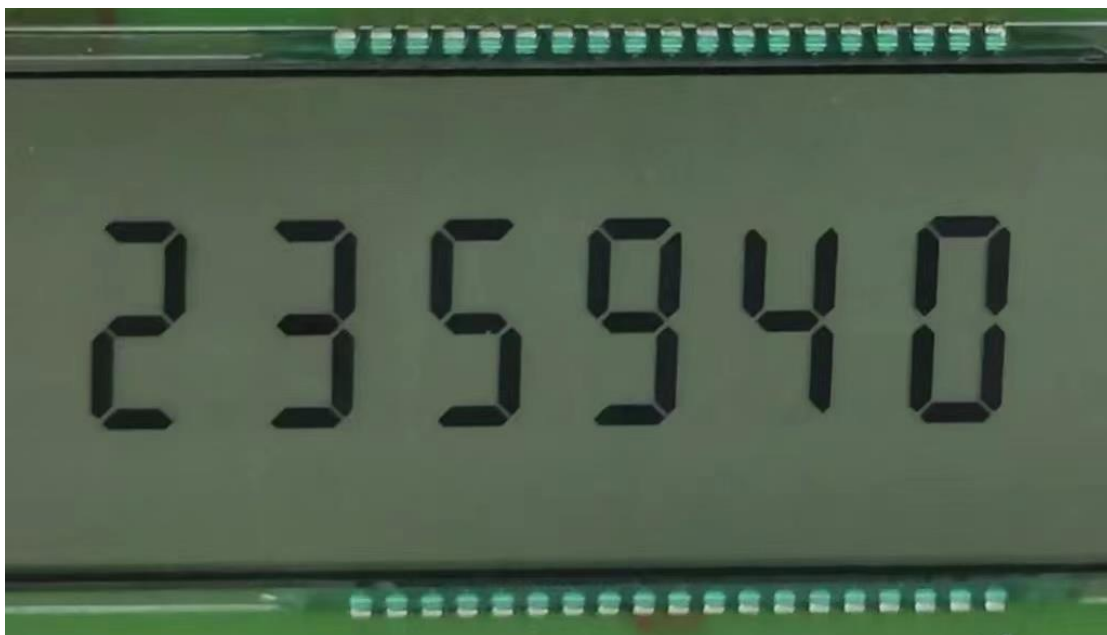


图 4. 显示时分秒

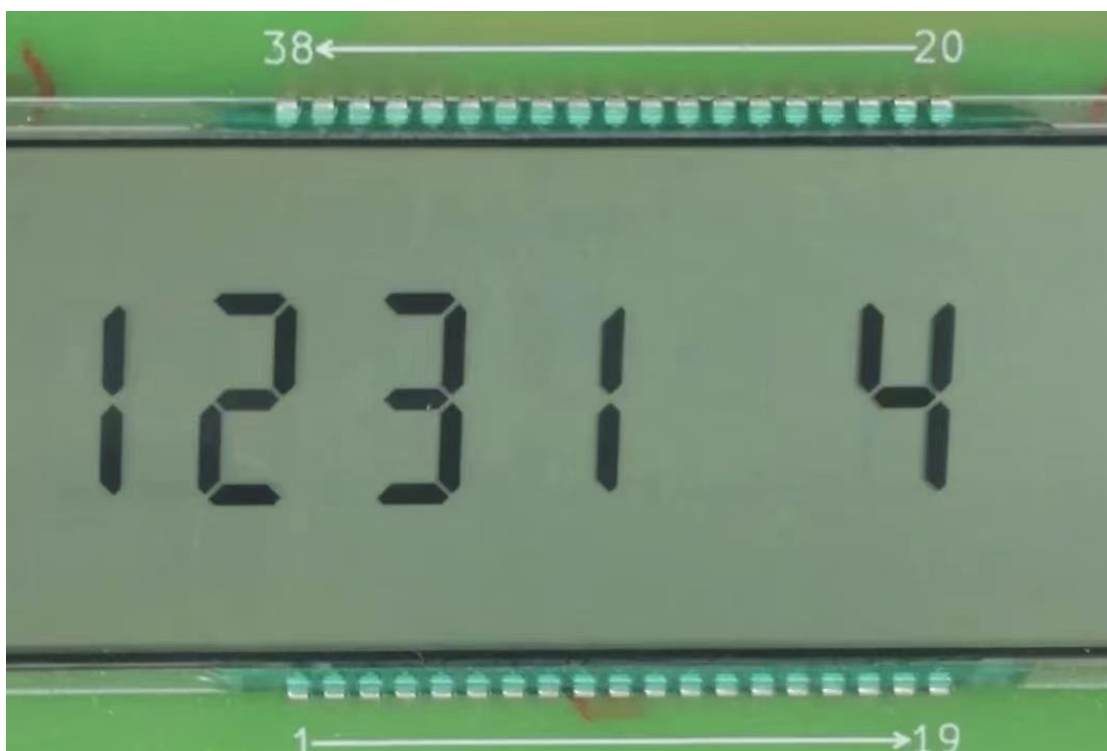


图 5. 显示月日星期

#### 四、开发问题简析：

开发阶段使用 RTC 功能要注意 RTC 的时钟源选择，为了获取更准确的计时效果，需要选择外部低频晶体作为时钟源，但晶体起振需要晶体与外围负载电容相匹配，因此要从晶体的相关资料中确认使用的晶体与实际的负载电容相匹配，才能保障晶体的有效起振。

#### 五、总结：

RTC 实时时钟显示功能的使用只要配合好 RTC 半秒中断和 LCD 显示模块就可以准确显示计时，更多的是对外围电路的要求。

附件：



示例程序： SD93F115-JBS\_TEST.zip