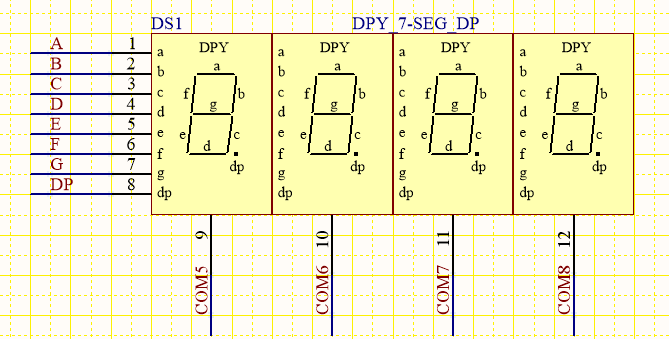
STM32 的实时时钟（RTC）是一个独立的定时器。STM32 的 RTC 模块拥有一组连续计数 的计数器，在相应软件配置下，可提供时钟日历的功能。修改计数器的值可以重新设置系统当 前的时间和日期。

1. Display\_set\_time函数

首先主函数中的Display\_set\_time函数是用数码管来显示实时时间的，在这个函数中调用了module\_74hc595.c中的数码管显示函数，具体见前面数码管的例程，我将要显示的时间分成个位和十位单独显示，比如data\_table[calendar.hour/10]显示十位，data\_table[calendar.hour%10]显示个位。



在选择16进制输入显示的数字是参看原理图。

2. hc595\_init，delay\_init这两个函数不过多介绍，详情见前面

3. RTC\_Init这个函数我们重点看下

\_calendar\_obj calendar;//时钟结构体，这个结构体在rtc.h文件中。

RTC 正常工作的一般配置步骤如下：

1. 使能电源时钟和备份区域时钟。

前面已经介绍了，我们要访问 RTC 和备份区域就必须先使能电源时钟和备份区域时钟。这 个通过 RCC\_APB1ENR 寄存器来设置。

1. 取消备份区写保护。

要向备份区域写入数据，就要先取消备份区域写保护（写保护在每次硬复位之后被使能）， 否则是无法向备份区域写入数据的。我们需要用到向备份区域写入一个字节，来标记时钟已经配置过了，这样避免每次复位之后重新配置时钟。

1. 复位备份区域，开启外部低速振荡器。

在取消备份区域写保护之后，我们可以先对这个区域复位，以清除前面的设置，当然这个 操作不要每次都执行，因为备份区域的复位将导致之前存在的数据丢失，所以要不要复位，要 看情况而定。然后我们使能外部低速振荡器，注意这里一般要先判断 RCC\_BDCR 的 LSERDY 位来确定低速振荡器已经就绪了才开始下面的操作。

1. 选择 RTC 时钟，并使能。

这里我们将通过 RCC\_BDCR 的 RTCSEL 来选择选择外部 LSI(32.768K 的外部晶振)作为 RTC 的时钟。然后通过 RTCEN 位使能 RTC 时钟。

1. 设置 RTC 的分频，以及配置 RTC 时钟。

在开启了 RTC 时钟之后，我们要做的就是设置 RTC 时钟的分频数，通过 RTC\_PRLH 和 RTC\_PRLL 来设置，然后等待 RTC 寄存器操作完成，并同步之后，设置秒钟中断。然后设置 RTC 的允许配置位（RTC\_CRH 的 CNF 位），设置时间（其实就是设置 RTC\_CNTH 和 RTC\_CNTL 两个寄存器）或者设置闹钟（设置 RTC\_ALRH 和 RTC\_ALRL 两个寄存器）。

1. 更新配置，设置 RTC 中断。

在设置完时钟之后，我们将配置更新，这里还是通过 RTC\_CRH 的 CNF 来实现。在这之后 我们在备份区域 BKP\_DR1 中写入 0X5050 代表我们已经初始化过时钟了，下次开机（或复位） 的时候，先读取 BKP\_DR1 的值，然后判断是否是 0X5050 来决定是不是要配置，避免重复配 置。接着我们配置 RTC 的秒钟中断，并进行分组。

1. 编写中断服务函数。

最后，我们要编写中断服务函数，在秒钟中断/闹钟中断产生的时候，读取当前的时间值， 并显示到 数码管上。 通过以上 7 个步骤，我们就完成了对 RTC 的配置，RTC 即可正常工作，而且这些操作不 是每次上电都必须执行的，可以视情况而定。当然，我们还需要设置时间、日期、闹钟等，这 些将在后面介绍。

在其中我们不是调用了 RTC\_Set(2015,1,14,17,42,55);

4.关于RTC\_Set

其实简单的理解你给他设定一个初始的时间，只不过系统通过秒数来计算的每一秒rtc中断函数，更新显示时间。在函数中我们调用了闰年计算的Is\_Leap\_Year函数，因为2.29这天只有在闰年才有，而中断函数每个一秒会自动执行的

5.关于rtc\_get

他的作用其实与RTC\_Set的部分作用相反的，函数其实就是将存储在秒钟寄存器 RTC->CNTH 和 RTC->CNTL 中的秒钟数据转换为真正 的时间和日期。而calendar 是我们在 rtc.h 里面将要定义 的一个时间结构体，用来存放时钟的年月日时分秒等信息，就是在Display\_set\_time中调用的字母。

6. RTC\_Get\_Week函数，了解就好，是用来计算星期几的，由于数码管有限我们只是显示了时分秒，所以这个函数从头到尾没有任何的作用，如果你以后想用lcd的话可以考录下。