APPLICATION NOTE

Real time clock with ST7 TIMER Output Compare

by 8-Bit Micro Application

INTRODUCTION

The purpose of this note is to present how to use the ST7 Timer output compare function. As an application example, this document presents a real time clock generation with second, minute and hour counters based on a fix basic time base.

1 GENERATING A TIME BASE WITH ST7 TIMER

The feature of the ST7 timer which allows to generate a fix time base is an output compare function.

TIMER CONFIGURATION AND INITIALIZATION

The described application is managed through output compare interrupt generation.

The ST7 timer contains two output compare functions which can be enabled only together through OCIE bit of the CR1 register. Then the Output Compare 1 (OC1) is used for the time base generation while the second one (OC2) is ignored generating an interrupt each free running counter loop.

To keep the minimum CPU load, the basic time base has to be a division of the second and the longest one. For these reason the maximum timer clock divider ratio (1/8) is selected in the CR2 register through CC1 and CC0 bits.

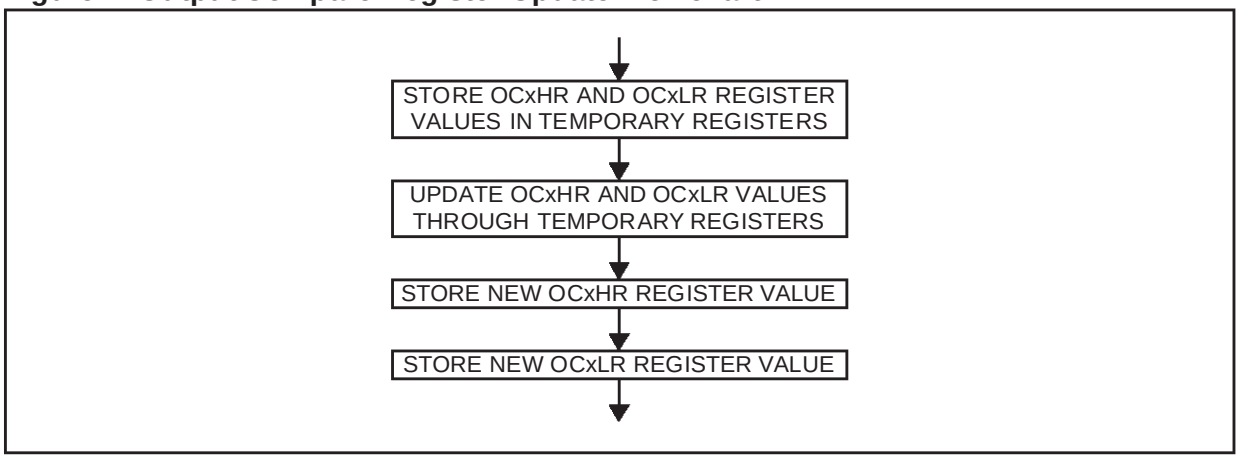
OUTPUT COMPARE UPDATE

To maintain a fix elapsed time between each output compare 1 interrupt generation, the Output compare registers (OC1HR, OC1LR) have to be updated with a fix offset time base at each interrupt.

$$(OCxHR,OCxLR) = (OCxHR,OCxLR) + (OCxHR,OCxLR)_{offset}$$

WARNING: As the ST7 can handle only 8-bit data and as the output compare function is inhibited between the write in the high (OCxHR) and the low (OCxLR) output compare registers, special care has to be taken in the update of the 16-bit compare register. A typical algorithm flowchart is described in the Figure 1.

Figure 1. Output Compare Register Update Flowchart



Considering that the f_{CPU} is 4-MHz, the Table 1 gives some basic possible time bases with the Output compare register offset value (OCxHR, OCxLR).

$$f_{timercnt} = \frac{f_{CPU}}{(cc1,cc0)} = \frac{4MHz}{8} = 500KHz \longrightarrow T_{timercnt} = 2\mu s$$

$$(\boldsymbol{OCxHR,OCxLR})_{offset} = \frac{\boldsymbol{TimeBase}}{\boldsymbol{T_{timecnt}}} = \frac{\boldsymbol{TimeBase}}{\boldsymbol{2\mu s}}$$

Table 1. Output Compare Register Correspondence with f_{CPU} at 4-MHz

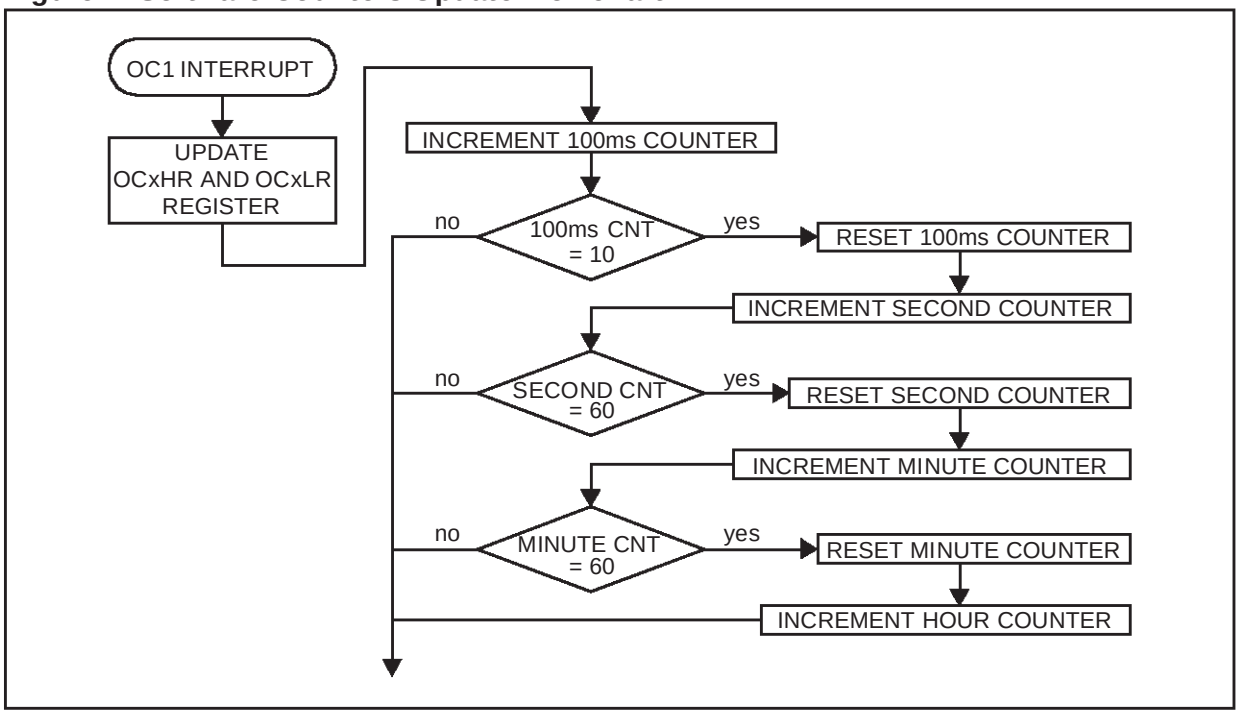
Time Base [ms]	2	10	20	50	100	131.07
OCxHR OCxLR offset [hex]	03E8	1388	2710	61A8	C350	FFFF

2 REAL TIME CLOCK APPLICATION

The real time clock application is based on four software counters (100ms, second, minute and hours) updated by the 100ms ST7 timer output compare 1 interrupt.

These four counters are coded on one byte each and are updated during the output compare 1 interrupt as shown in Figure 2.

Figure 2. Software Counters Update Flowchart

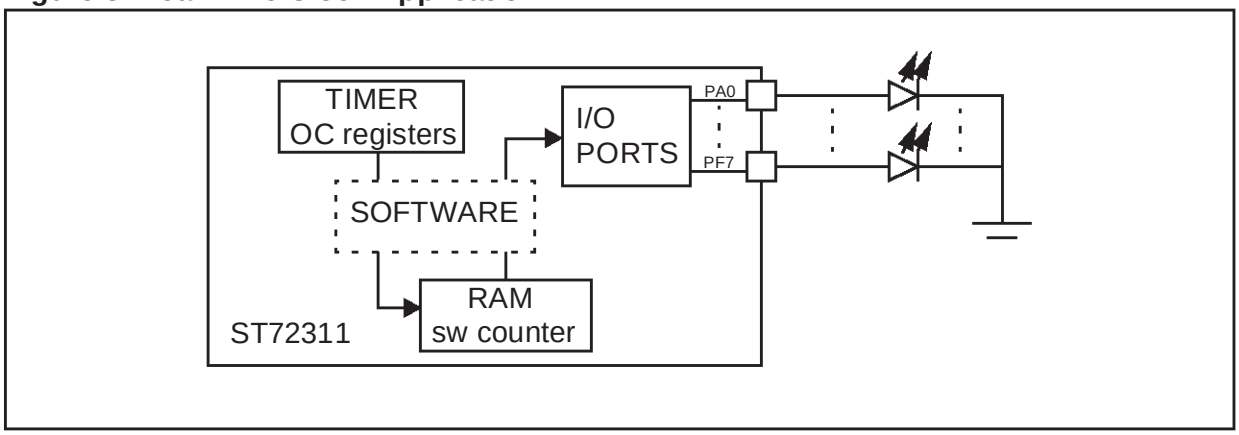


HARDWARE CONFIGURATION

The real time clock application hardware is made of a ST72311 microcontroller timer which generates a 100ms time base through the output compare 1 of the timer B. Two of the software counters (second and minute ones) are output from the micro through I/O ports (PA and PF) and displayed on LED as described in Figure 3.

The selected cristal for this example is a 8-MHz one to work with f_{CPU}=4-MHz.

Figure 3. Real Time Clock Application



3 SOFTWARE

The assembly code given below is for guidance only. For missing label declaration please refer to the register label description of the datasheet or the ST web software library ("ST72311.inc" file...).

```
st7/
;****************** (c) 1998 STMicroelectronics **********************
; PROJECT  : REAL TIME CLOCK WITH ST7 TIMER OUTPUT COMPARE DEMO SYSTEM
; COMPILER : ST7 ASSEMBLY CHAIN
; MODULE  :  tim_rtc.asm
; CREATION DATE : 14/04/98
; AUTHOR : 8-bit Micro Application Team
;-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
;   THE SOFTWARE INCLUDED IN THIS FILE IS FOR GUIDANCE ONLY. STMicroelectronics
;   SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL
;   DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM USE OF THIS SOFTWARE.
;-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
; DESCRIPTION :  ST7 Timer OC software driver for Real Time Clock application.
;******************************************************************************

       TITLE  "TIM_RTC.ASM"
       MOTOROLA
       #INCLUDE "ST72311.inc"     ; ST72311 registers and memory mapping file

;******************************************************************************
;    Macro definitions
;******************************************************************************

   #define OCF1   6    ; Timer status register Output compare flag 1 position.

;******************************************************************************
;    RAM SEGMENT
;******************************************************************************

       BYTES                          ; following addresses are 8 bit length
       segment byte at 80-FF 'ram0'

.t_hour     DS.B 1                          ; Real time clock HOUR  counter.
.t_min      DS.B 1                         ; Real time clock MINUTE counter.
.t_sec      DS.B 1                         ; Real time clock SECOND counter.
.t_100ms    DS.B 1                  ; Counter up to 10 interrupts OC1 = 1sec.
```

```
;*******************************************************************************
;    ROM SEGMENT
;*******************************************************************************

       WORDS
       segment 'rom'

; ------------------------------------------------------------------------------
; ROUTINE NAME : TRtc_Init
; INPUT/OUTPUT : None.
; DESCRIPTION  : TIMER initialization for PWM mode.
; ------------------------------------------------------------------------------
.TRtc_Init  ; IN: None / OUT: None.
       LD   A,#$40                      ; Output compare interrupt enable.
       LD   TBCR1,A
       LD   A,#$08              ; Internal clock divided by 8 => CC1=1 CC0=0.
       LD   TBCR2,A
       CLR  A                                          ; RTC Counter reset.
       LD   t_hour,A
       LD   t_min,A
       LD   t_sec,A
       LD   t_100ms,A
       RIM                   ; Enable interrupt: launch the real time clock.
       RET

;-------------------------------------------------------------------------------
; ROUTINE NAME : TRtc_CntUpdate
; INPUT/OUTPUT : None.
; DESCRIPTION  : Update the main loop timer TIMER 2 counters.
;-------------------------------------------------------------------------------
.TRtc_CntUpdate ; IN: None / OUT: None.
       INC  t_100ms     ; Increment the OC1 counter.
       LD   A,t_100ms
       CP   A,#10
       JRNE cntend
       CLR  t_100ms     ; 1 second is reached.
       INC  t_sec
       LD   A,t_sec
       CP   A,#60
       JRNE cntend
       CLR  t_sec       ; 1 minute is reached.
       INC  t_min
       LD   A,t_min
       CP   A,#60
       JRNE cntend
       CLR  t_min       ; 1 hour is reached.
       INC  t_hour
.cntendRET
```

```
; ------------------------------------------------------------------------------
; ROUTINE NAME : TRtc_Int
; INPUT/OUTPUT : None
; DESCRIPTION  : Interrupt routine for the ST7 TIMER wit fCPU=4MHz.
; COMMENTS     : Only Output compare interrupt is taken into account.
;-------------------------------------------------------------------------------
.TRtc_Int   ; IN: None / OUT: None.
       BTJF TBSR,#OCF1,intoc2
       ; Interrupt generated by Output Compare 1.
       ; INTERRUPT each 100ms (OC1 = 0xC350 with internal clock div by 8).
       LD   A,TBOC1LR                          ; TmpOc = TIM2_OC1 + 0xC350.
       ADD  A,#$50                                     ; Add LSB value.
       LD   Y,A
       LD   A,TBOC1HR
       ADC  A,#$C3                                     ; Add MSB value.
       LD   TBOC1HR,A     ; Update TBOC1 registers with new calculated value.
       LD   A,Y
       LD   TBOC1LR,A
       CALL TRtc_CntUpdate         ; Real time clock software counter update.
.intoc2LD   A,TBOC2LR           ; Access to OC2 low byte in order to reset OCF2
       IRET                      ; and pass a possible output compare 2 event.


;    ****************************************
;    *     MAIN-ROUTINES SECTION      *
;    ****************************************

.main ; Initialization ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       CLR  PAOR
       CLR  PFOR
       LD   A,#$FF         ; Configure PA and PF IO ports in output push-pull.
       LD   PADDR,A
       LD   PFDDR,A
       CALL TRtc_Init                             ; Init TIMER peripheral.
   ; ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
.loop  LD   A,t_sec                          ; Exit second data on PF IO port.
       LD   PFDR,A
       LD   A,t_min                          ; Exit minute data on PA IO port.
       LD   PADR,A
       JRA  loop                                    ; Infinity main loop.
; ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
;       ******************************************
;       *    INTERRUPT SUB-ROUTINES LIBRARY SECTION*
;       ******************************************

.dummy_rt   iret      ; Empty subroutine. Go back to main (iret instruction).

        segment 'vectit'
              DC.W   dummy_rt          ; FFF0-FFF1h location
              DC.W   dummy_rt          ; FFF2-FFF3h location
              DC.W   dummy_rt          ; FFF4-FFF5h location
sci_it:       DC.W   dummy_rt          ; FFE6-FFE7h location
timb_it:      DC.W   TRtc_Int          ; FFE8-FFE9h location
tima_it:      DC.W   dummy_rt          ; FFEA-FFEBh location
spi_it:       DC.W   dummy_rt          ; FFEC-FFEDh location
              DC.W   dummy_rt          ; FFEE-FFEFh location
ext3_it:      DC.W   dummy_rt          ; FFF0-FFF1h location
ext2_it:      DC.W   dummy_rt          ; FFF2-FFF3h location
ext1_it:      DC.W   dummy_rt          ; FFF4-FFF5h location
ext0_it:      DC.W   dummy_rt          ; FFF6-FFF7h location
              DC.W   dummy_rt          ; FFF8-FFF9h location
              DC.W   dummy_rt          ; FFFA-FFFBh location
softit:       DC.W   dummy_rt          ; FFFC-FFFDh location
reset:        DC.W   main              ; FFFE-FFFEh location

        END
;*** (c) 1998 STMicroelectronics ****************** END OF FILE ****
```

THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNEXION WITH THEIR PRODUCTS.