Lecture-8: *Real-Time Clock*

By

Syed Masud Mahmud, Ph.D.

Copyright © 2002 by Syed Masud Mahmud
An Algorithm for a Real-Time Clock

• We can write a program for a Real-Time Clock using an Output Compare function of the timer.
• We can maintain the following locations in memory to keep track of the time.
  - SEC
  - MIN
  - HOUR
  - AMPM
An Algorithm for a Real-Time Clock

- We will write a routine called the SEC_ROUTINE.
- This routine will be called once every second.
- The SEC_ROUTINE will do the following operations:
  1. SEC ← SEC + 1
  2. IF SEC = 60
     THEN SEC ← 0 and MIN ← MIN + 1
  3. IF MIN = 60
     THEN MIN ← 0 and HOUR ← HOUR + 1
An Algorithm for a Real-Time Clock (contd.)

4. IF HOUR = 12
   THEN Toggle AMPM
       (i.e. If AMPM = 0
        Then AMPM ← 1 Else AMPM ← 0)

5. IF HOUR = 13 THEN HOUR ← 1
Template for a Real-Time Clock

*-------------------------------------*
* TEMPLATE FOR A REAL-TIME CLOCK     *
*-------------------------------------*
* Author: Syed Masud Mahmud          *
*-------------------------------------*
LCDBAS EQU $B5F0                      * LCD port address
*-------------------------------------*
REGBAS EQU $1000
PORTA EQU 0
TCNT EQU $0E
TOC3 EQU $1A
TMSK1 EQU $22
TFLG1 EQU $23
TWENTYFIVE_MS EQU 50000
ONE_SEC EQU 40
**Template for a Real-Time Clock (contd.)**

<table>
<thead>
<tr>
<th>ORG</th>
<th>RMB</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC3_CNT</td>
<td>1</td>
<td>$200</td>
</tr>
<tr>
<td>CHAR1</td>
<td>1</td>
<td>* Locations for</td>
</tr>
<tr>
<td>CHAR2</td>
<td>1</td>
<td>* keeping three</td>
</tr>
<tr>
<td>CHAR3</td>
<td>1</td>
<td>* ASCII codes</td>
</tr>
<tr>
<td>HOUR</td>
<td>1</td>
<td>* location for HOUR</td>
</tr>
<tr>
<td>MIN</td>
<td>1</td>
<td>* location for MIN</td>
</tr>
<tr>
<td>SEC</td>
<td>1</td>
<td>* location for SEC</td>
</tr>
<tr>
<td>AMPM</td>
<td>1</td>
<td>* location for AMPM</td>
</tr>
</tbody>
</table>

* *OC3 VECTOR*

<table>
<thead>
<tr>
<th>ORG</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$00D9</td>
<td>OC3 Vector for running</td>
</tr>
<tr>
<td>JMP OC3SERV</td>
<td>from RAM</td>
</tr>
</tbody>
</table>
*--- MAIN PROGRAM STARTS FROM HERE ---*

ORG $1040  * start code at $1040
JMP MAIN   * jump over data to start of program

MESS
FCC 'TIME :: '
FCB 0

AM
FCC ' AM'
FCB 0

PM
FCC ' PM'
FCB 0

*--- Main program starts from here ---*

MAIN SEI          * Disable interrupts
JSR LCDSET       * Setup LCD
* Load locations HOUR, MIN, SEC with some initial values.
* 
LDAA  #12  
STAA  HOUR  
LDAA  #00  
STAA  MIN  
LDAA  #00  
STAA  SEC  
LDAA  #0  * Initial Time is  
STAA  AMPM  * 12:00:00 AM  

CLR   OC3_CNT  
LDX   #REGBAS   ; X = $1000 = PortA Addr.  
LDD   TCNT,X    ; Read timer  
ADDD  #TWENTYFIVE_MS ; Add value for 20 ms  
STD   TOC3,X    ; delay and store in TOC3  
BSET  TMSK1,X $20 ; Set OC3I bit  
BCLR  TFLG1,X $DF ; Clear OC3F bit  
CLI   ; Clear I-bit of CCR
Template for a Real-Time Clock (contd.)

```
LOOP
*---------------------------------------------------------------------*
*  HERE, WRITE THE CODE FOR CHECKING THE SWITCHES FOR                *
*  SETTING UP THE CLOCK                                              *
*---------------------------------------------------------------------*
*---------------------------------------------------------------------*
BRA LOOP ; Repeat
*---------------------------------------------------------------------*
*---------------------------------------------------------------------*
OC3 SERVICE ROUTINE
*---------------------------------------------------------------------*
OC3SERV LDD TOC3,X ; Read TOC3 and then
ADDD #TWENTYFIVE_MS ; add value for another
STD TOC3,X ; 20ms delay
INC OC3_CNT
LDAA #ONE_SEC
CMPA OC3_CNT
BNE OC3DONE
JSR SEC_ROUTINE
CLR OC3_CNT
OC3DONE BCLR TFLG1,X $DF ; Clear OC3F bit
RTI
```
* The processor will come to this routine once every second.

SEC_ROUTINE

PSHX
INC SEC

* HERE, WRITE THE CODE FOR REAL-TIME CLOCK

* 

JSR ROW2 * Move the cursor to Row-2
LDX #MESS
JSR PRINT * Display 'TIME :: '
JSR DSPTIME * Display TIME on the LCD
PULX
RTS