OVERVIEW

This 99 minute timer replaces the electro-mechanical timer. The hardware is made simple and economical. The timer is very simple to operate. Two 7-segment displays are provided to display minutes from 0 to 99. To simplify operation, only two push-on switches are provided to set minutes. Each switch is dedicated to each display.

“TENS OF MIN SW” increments tens of minutes display from 0 through 9 at each push and “UNITS OF MIN SW” increments units of minutes displays from 0 through 9 at each push. Debouncing of switches is taken care by software.

APPLICATION OPERATION

The TMR0 prescaler is set to x16. Software checks TMR0 for 250 rollovers, which happens every 4 msec. 4 msec is counted 25 times to generate 100 msec, from which seconds and minutes are generated.

Every 100 msec the keyboard is read and the required minutes display (TENS or UNITS) is incremented at the rate of 2 Hz.

If the displayed minutes is equal to 0 then relay is put off and UNITS DISPLAY’s decimal point is not flashing to indicate the timer is off.

If displayed minutes is not equal to 0, then a relay is turned on and UNITS DISPLAY’s decimal point is flashed to indicate the timer is active. During this mode the display decrements every minute till it reaches 0, where the relay is put off.

Note 1: Total instruction time [one loop]: Approximately 375 µsec, maximum.

2: Set Processor configuration word as 0000 0000 1010b.
   a) MCLR tied to VDD (internally).
   b) Code protection off.
   c) WDT disabled.
   d) Internal RC oscillator.

Author: Milind Hareshwar Patil
Tata Institute Of Fundamental Research
Mumbai, Maharashtra
India
email: milindhp@tifrvax.tifr.res.in

99 Minute Timer

Electromechanical Timer Replacement
FIGURE 1: SCHEMATIC

MORTICHIP TOOLS USED

Assembler/Comilier Version

MPASM 1.50 released.
APPENDIX A: SOURCE CODE

; define ram ports and data ; ***************************** ;
DEFINE PORT
;
; DEFINE PORTS
 ***********
gp0 equ 0
gp1 equ 1
gp2 equ 2
gp3 equ 3
gp4 equ 4
gp5 equ 5

; DISPLAY PORT
 ***********
disp_data equ gp0
disp_clk equ gp1
disp_strobe equ gp2

; KEYBOARD PORT
 *************
units_key equ gp3	units_key equ gp3
tens_key equ gp4
tens_key equ gp4

; RELAY PORT
 **********
relay equ gp5

; DEFINE RAM
******************************************************
units equ 08h ; minutes unit display
tens equ units+1 ; minutes tens display
counter_4ms equ tens+1 ; incremented every 4 msec
counter_100ms equ counter_4ms+1 ; incremented every 100 msec
seconds equ counter_100ms+1
oldkey equ seconds+1 ; for key debouncing
newkey equ oldkey+1 ; ------- ,, -------
key equ newkey+1 ; key pressed data
digit_inc equ key+1 ; inc units display if bo = 1
digit_inc equ key+1

tmr_comp equ digit_inc+1 ; tmr0 comparator register
scrtch0 equ tmr_comp+1
scrtch1 equ scrtch0+1
scrtch2 equ scrtch1+1

; DEFINE FLAGS AND BITS
******************************************************
b0 equ 0
b1 equ 1
b2 equ 2
b3 equ 3
b4 equ 4
b5 equ 5
b6 equ 6
b7 equ 7

; KEY
***
equ b0
equ b1
equ b2
units_key equ b3
tens_key equ b4

; digit_inc
**********
units_inc equ b0
tens_inc equ b1

; equ b3
; equ b4
; equ b5
; equ b6
; equ b7

;******************************************************************************
#define data_hi    bsf     GPIO, disp_data
#define data_lo    bcf     GPIO, disp_data
#define clk_hi     bsf     GPIO, disp_clk
#define clk_lo     bcf     GPIO, disp_clk
#define strobe_hi  bsf     GPIO, disp_strobe
#define strobe_lo  bcf     GPIO, disp_strobe
#define relay_on   bsf     GPIO, relay
#define relay_off  bcf     GPIO, relay
;************* EOF DEF_RAM.ASM *************

;************** EOF SUBS.ASM **************

; read key port in key
read_keys       movf    newkey, w       ;debouncing taken care by s/w
       movwf   oldkey
       movf    GPIO, w         ;read newkeys
       andlw   00011000b       ;gp3, gp4 are key ports
       movwf   newkey
       comf    newkey, f       ;complement since active low
       andwf   oldkey, w       ;key=oldkey AND compl(newkey)
       movwf   key
       ;indicate to display routine units or tens key pressed
       btfsc   key, units_key
       bsf     digit_inc, units_inc
       btfsc   key, tens_key
       bsf     digit_inc, tens_inc
       return

;*************** EOF SUBS.ASM **************************

;init_ports      movlw   0               ;all lo
       movwf   GPIO
       movlw   00011000b       ;port g0-g2,g5 o/p & gp3,gp4 i/p
       tris    GPIO
       clrf    TMR0            ;clr tmr0 & prescaler
       movlw   11000011b       ;tmr0 enable with 1:16 prescaler
     option
       return

;*************** EOF TABLES.ASM **************************

;7 segments decoded data. lo is segment on & hi is segment off
get_seg         addwf   PCL, f
; ABCDEFGP      ;P is decimal point
       retlw   00000011b       ;0
       retlw   10011111b       ;1
       retlw   000101101b      ;2
       retlw   000011011b      ;3
       retlw   10011001b       ;4
       retlw   01001001b       ;5
       retlw   01000001b       ;6
       retlw   00011111b       ;7
       retlw   00000001b       ;8
       retlw   00011001b       ;9
dec_pt          equ     11111110b       ;decimal point bit

;*************** EOF TABLES.ASM **************************
; TIMER.ASM

; ********

; [ milindhp@tifrvax.tifr.res.in ]

; Set Processor configuration word as = 0000 0000 1010 b.
; a] -MCLR tied to VDD (internally).
; b] Code protection off.
; c] WDT disabled.
; d] Internal RC oscillator [4 MHZ].

list p=12c508, r=dec
include "d:\pic\mpasm\p12c508.inc"

;define ram
include "def_ram.asm"

;processor start
org 0
goto start

;define legends and table
include "tables.asm"

;subroutines
include "subs.asm"

;initialize and start
start
    clrf key
    clrf oldkey
    clrf newkey
    clrf digit_inc
    clrf units
    clrf tens
    clrf counter_4ms
    clrf counter_100ms
    clrf seconds
    movlw 250
    ; 4000us tmr0 (1:16 prescaler * 250)
    movwf tmr_comp
    call init_ports
    ; init ports & timer
    movf tmr_comp, w
    ; is tmr0 = tmr_comp (4 msec over?)
    xorwf TMR0, w
    btfss STATUS, Z
    ; skip if = 250
    goto main
    movlw 250
    addwf tmr_comp, f
    incf counter_4ms, f
    movlw 25
    xorwf counter_4ms, w
    ; is 4ms * 25 = 100ms over?
    btfss STATUS, Z
    ; skip if 100ms
    goto main21
    clrf counter_4ms
    ; 1 sec over
    incf seconds, f
    movlw 60
    xorwf seconds, w
    ; is 1 minute over?
    btfss STATUS, Z
    ; skip if 1 min over
    goto main21

; *********** 100 MSEC OVER ********************************************

; *********** 1 MINUTE OVER ********************************************

; process relay and display
; if tens & units both = 0 then rly off and out
; else relay on & decrement display & out
movf tens, w
iorwf units, w
btfss STATUS, Z
goto main1
relay_off
goto main21
main1
    relay_on
    movf units, f
    btfss STATUS, Z
    goto main2
    movlw 9
    movwf units
    decf tens, f
    goto main21

;************************** EXECUTED EVERY 100 MSEC ***************
main21
call read_keys

;** DISPLAY ROUTINE TO BE EXECUTED EVERY 500 MSEC **
;process display every 500 msec
;if unit key pressed increment unit display
;if tens key pressed increment tens display
;flash unit display's dec. pt. every second if units & tens
;not equal to zero i.e. indicate timer is active
;execute display routine in counter_100ms = 0 or 5
movf counter_100ms, w
    btfsc STATUS, Z
    goto main3
    xorlw 5
    btfss STATUS, Z
    goto main

;if units_key pressed then inc units display between 0 thr. 9
;if tens_key pressed then inc tens display between 0 thr. 9
main3
    btfss digit_inc, units_inc
    goto main4
    bcf digit_inc, units_inc
;inc units display. if units = 10 then units = 0
    incf units, f
    movlw 10
    xorwf units, w
    btfsc STATUS, Z
    ;skip if not equal
    clrf units

main4
    btfss digit_inc, tens_inc
    goto main5
    bcf digit_inc, tens_inc
;inc tens display. if tens = 10 then tens = 0
    incf tens, f
    movlw 10
    xorwf tens, w
    btfsc STATUS, Z
    ;skip if not equal
    clrf tens

;convert decimal in units & tens to decoded segment data
;for led display in scrtch1 & 2 for transmission
main5
    movf tens, w
    call get_seg
    movwf scrtch1
    movf units, w
    call get_seg
    movwf scrtch2

;flash (toggle) decimal point every 500ms if timer active
;i.e. if timer not equal to zero i.e. relay on, then flash
;decimal point else, do not to indicate relay off
movf tens, w
    iorwf units, w
    btfsc STATUS, Z
    ;skip to flash dp
    goto main6
    ;jmp if 0 to no flash
    movf counter_100ms, f
    btfss STATUS, Z
    ;toggle-on dec pt if counter_100ms = 0
    goto main6
    ;else jmp out to toggle-off

;flash dp of units display
movlw dec_pt
    andwf scrtch2, f
; transmit data from scrtch1 thr scrtch1 to display circuit (msb first).
main6     movlw    16              ; no of bit to tx
          movwf    scrtch0
main61    rlf     scrtch2, f      ; check msb
          rlf     scrtch1, f
          btfsc   STATUS, C       ; data hi if cy=1 else lo
          goto    main62
          data_lo
          goto    main63
main62    data_hi
main63    nop                     ; delay
          nop
          clk_hi                ; toggle clk to push data
          nop
          clk_lo
          nop
          decfsz   scrtch0, f      ; next bit
          goto    main61
          strobe_hi             ; strobe data nop
          nop
          strobe_lo
          data_lo               ; leave data lo
          goto    main
end
; ************* EOF MAIN.ASM *******************************