```
;*************************************************************
;* CMPEN 472, 2022 Spring
;* Homework 10: Timer Interrupt Sample Program,
;* MC9S12C128 Program (set to MC9S12C32 for Simulation/Debug)
;* CodeWarrior Simulator/Debug edition, not for CSM-12C128 board
;* Nov.   01, 2021 Kyusun Choi
;* March  27, 2022 Kyusun Choi
;* Nov.   07, 2022 Kyusun Choi
;*
;* This program is a 1024 data transfer program running on the
;*   CodeWarrior Debugger/Simulator as follows:
;*   1. Program starts with print messages on the simulator Terminal,
;*      an intro message at 1.5M baud (this program will not work
;*      on the CSM-12C128 board - 1.5M baud too fast).
;*   2. Then user may hit any key, it's a typewriter program at 1.5M baud.
;*      But hitting the Enter key will terminate the typewriter mode with
;*      the instruction message print.
;*   3. Two messages are (1) start terminal data capture into a file and
;*      (2) hit Enter key for the 1024 data transfer to begin.
;*   4. At this time, user setup the Terminal Output file, data capture to a file.
;*   5. User hits an Enter key to send 1024 data, to the Terminal and
;*      the data saved in to a file named RxData3.txt  which may be looked at
;*      or plotted using Excel sheet.
;*   6. User may repeat the step 3 above as many times as he/she like.
;*      User plots or prints the data to verify the correct data transmit.
;*
;* We assumed 24MHz bus clock and 4MHz external resonator clock frequency.
;*
;*************************************************************
;*************************************************************

; export symbols - program starting point
            XDEF        Entry   ; export 'Entry' symbol
            ABSENTRY    Entry   ; for assembly entry point

; include derivative specific macros
PORTB       EQU         $0001
DDRB        EQU         $0003

SCIBDH      EQU         $00C8   ; Serial port (SCI) Baud Register H
SCIBDL      EQU         $00C9   ; Serial port (SCI) Baud Register L
SCICR2      EQU         $00CB   ; Serial port (SCI) Control Register 2
SCISR1      EQU         $00CC   ; Serial port (SCI) Status Register 1
SCIDRL      EQU         $00CF   ; Serial port (SCI) Data Register

TIOS        EQU         $0040   ; Timer Input Capture (IC) or Output Compare (OC) select
TIE         EQU         $004C   ; Timer interrupt enable register
TCNTH       EQU         $0044   ; Timer free runing main counter
TSCR1       EQU         $0046   ; Timer system control 1
TSCR2       EQU         $004D   ; Timer system control 2
TFLG1       EQU         $004E   ; Timer interrupt flag 1
TC6H        EQU         $005C   ; Timer channel 2 register

CR          equ         $0d     ; carriage return, ASCII 'Return' key
LF          equ         $0a     ; line feed, ASCII 'next line' character

DATAmax     equ         1024    ; Data count maximum, 1024 constant

;*************************************************************
; variable/data section
            ORG    $3000        ; RAMStart defined as $3000
                                ; in MC9S12C128 chip

ctr125u     DS.W   1            ; 16bit interrupt counter for 125 uSec. of time
```

```
BUF         DS.B   6           ; character buffer for a 16bit number in decimal ASCII
CTR         DS.B   1           ; character buffer fill count

msg1        DC.B   'Hello, this is 1024 data transmit program.', $00
msg2        DC.B   'When ready, hit Enter key.', $00
;*          more text messages at the End of this program

;***************************************************
; interrupt vector section

            ORG     $FFE2      ; Timer channel 6 interrupt vector setup, on simulator
            DC.W    oc6isr

;***************************************************
; code section

            ORG    $3100
Entry
            LDS    #Entry      ; initialize the stack pointer

            LDAA   #%11111111  ; Set PORTB bit 0,1,2,3,4,5,6,7
            STAA   DDRB        ; as output
            LDAA   #%00000000  ; Clear PORTB bit 0,1,2,3,4,5,6,7
            STAA   PORTB       ; Clear all bits of PORTB, initialize

            ldaa   #$0C        ; Enable SCI port Tx and Rx units
            staa   SCICR2      ; disable SCI interrupts

            ldd    #$0001      ; Set SCI Baud Register = $0001 => 1.5M baud at 24MHz (for
simulation)
;            ldd    #$0002       ; Set SCI Baud Register = $0002 => 750K baud at 24MHz
;            ldd    #$000D       ; Set SCI Baud Register = $000D => 115200 baud at 24MHz
;            ldd    #$009C       ; Set SCI Baud Register = $009C => 9600 baud at 24MHz
            std    SCIBDH      ; SCI port baud rate change

            ldx     #msg1           ; print the first message, '1024 data transmit'
            jsr     printmsg
            jsr     nextline

            ldx     #msg2           ; print the second message, user instruction,
            jsr     printmsg        ;   hit 'Enter'
            jsr     nextline

mloop1
            jsr     getchar
            cmpa    #0
            beq     mloop1
            jsr     putchar         ; type writer, with echo print
            cmpa    #CR
            bne     mloop1          ; if Enter/Return key is pressed, move the

            ldaa    #LF             ; cursor to next line
            jsr     putchar

            ldx     #msg3           ; print '> Set Terminal save file RxData3.txt'
            jsr     printmsg
            jsr     nextline

            ldx     #msg4           ; print '> Press Enter/Return key to start sawtooth
wave'
            jsr     printmsg
            jsr     nextline

            jsr     delay1ms        ; flush out SCI serial port
```

```
                                    ; wait to finish sending last characters

mloop2
            jsr     getchar
            cmpa    #0
            beq     mloop2
            cmpa    #CR
            bne     mloop2          ; if Enter/Return key is pressed, move the

            jsr     nextline
            jsr     nextline
            jsr     delay1ms        ; flush out SCI serial port
                                    ; wait to finish sending last characters

            ldx     #0              ; Enter/Return key hit
            stx     ctr125u
            jsr     StartTimer6oc

            CLI                     ; Interrupt enable, for Timer OC6 interrupt start


loop1024
            ldd     ctr125u
            cpd     #DATAmax        ; 1024 bytes will be sent, the receiver at Windows PC
            bhs     loopTxON        ;   will only take 1024 bytes.
            bra     loop1024        ; set Terminal Cache Size to 10000 lines, update from
1000 lines

loopTxON
            LDAA    #%00000000
            STAA    TIE              ; disable OC6 interrupt

            jsr     nextline
            jsr     nextline

            ldx     #msg5           ; print '> Done!  Close Output file.'
            jsr     printmsg
            jsr     nextline

            ldx     #msg6           ; print '> Ready for next data transmission'
            jsr     printmsg
            jsr     nextline

            BRA     mloop2


;subroutine section below

;***********Timer OC6 interrupt service routine***************
oc6isr
            ldd   #3000             ; 125usec with (24MHz/1 clock)
            addd  TC6H              ;    for next interrupt
            std   TC6H              ;
            bset  TFLG1,%01000000   ; clear timer CH6 interrupt flag, not needed if fast
clear enabled
            ldd   ctr125u
            ldx   ctr125u
            inx                     ; update OC6 (125usec) interrupt counter
            stx   ctr125u
            clra                    ;   print ctr125u, only the last byte
            jsr   pnum10            ;   to make the file RxData3.txt with exactly 1024 data
oc2done     RTI
;***********end of Timer OC6 interrupt service routine********

;***************StartTimer6oc**************************
```

```
;* Program: Start the timer interrupt, timer channel 6 output compare
;* Input:   Constants - channel 6 output compare, 125usec at 24MHz
;* Output:  None, only the timer interrupt
;* Registers modified: D used and CCR modified
;* Algorithm:
;             initialize TIOS, TIE, TSCR1, TSCR2, TC2H, and TFLG1
;*********************************************
StartTimer6oc
            PSHD
            LDAA   #%01000000
            STAA   TIOS              ; set CH6 Output Compare
            STAA   TIE               ; set CH6 interrupt Enable
            LDAA   #%10000000        ; enable timer, Fast Flag Clear not set
            STAA   TSCR1
            LDAA   #%00000000        ; TOI Off, TCRE Off, TCLK = BCLK/1
            STAA   TSCR2             ;   not needed if started from reset

            LDD    #3000            ; 125usec with (24MHz/1 clock)
            ADDD   TCNTH            ;    for first interrupt
            STD    TC6H             ;

            BSET   TFLG1,%01000000   ; initial Timer CH6 interrupt flag Clear, not needed if
fast clear set
            LDAA   #%01000000
            STAA   TIE               ; set CH6 interrupt Enable
            PULD
            RTS
;***************end of StartTimer2oc*****************


;***********pnum10****************************
;* Program: print a word (16bit) in decimal to SCI port
;* Input:   Register D contains a 16 bit number to print in decimal number
;* Output:  decimal number printed on the terminal connected to SCI port
;*
;* Registers modified: CCR
;* Algorithm:
;     Keep divide number by 10 and keep the remainders
;     Then send it out to SCI port
;  Need memory location for counter CTR and buffer BUF(6 byte max)
;*********************************************
pnum10          pshd                    ;Save registers
                pshx
                pshy
                clr     CTR             ; clear character count of an 8 bit number

                ldy     #BUF
pnum10p1        ldx     #10
                idiv
                beq     pnum10p2
                stab    1,y+
                inc     CTR
                tfr     x,d
                bra     pnum10p1

pnum10p2        stab    1,y+
                inc     CTR
;---------------------------------------

pnum10p3        ldaa    #$30
                adda    1,-y
                jsr     putchar
                dec     CTR
                bne     pnum10p3
                jsr     nextline
```

```
                puly
                pulx
                puld
                rts
;***********end of pnum10*******************

;***********printmsg****************************
;* Program: Output character string to SCI port, print message
;* Input:   Register X points to ASCII characters in memory
;* Output:  message printed on the terminal connected to SCI port
;*
;* Registers modified: CCR
;* Algorithm:
;     Pick up 1 byte from memory where X register is pointing
;     Send it out to SCI port
;     Update X register to point to the next byte
;     Repeat until the byte data $00 is encountered
;       (String is terminated with NULL=$00)
;**********************************************
NULL            equ     $00
printmsg        psha                    ;Save registers
                pshx
printmsgloop    ldaa    1,X+            ;pick up an ASCII character from string
                                        ;   pointed by X register
                                        ;then update the X register to point to
                                        ;   the next byte
                cmpa    #NULL
                beq     printmsgdone    ;end of strint yet?
                bsr     putchar         ;if not, print character and do next
                bra     printmsgloop
printmsgdone    pulx
                pula
                rts
;***********end of printmsg*******************

;***************putchar************************
;* Program: Send one character to SCI port, terminal
;* Input:   Accumulator A contains an ASCII character, 8bit
;* Output:  Send one character to SCI port, terminal
;* Registers modified: CCR
;* Algorithm:
;    Wait for transmit buffer become empty
;      Transmit buffer empty is indicated by TDRE bit
;      TDRE = 1 : empty - Transmit Data Register Empty, ready to transmit
;      TDRE = 0 : not empty, transmission in progress
;**********************************************
putchar     brclr SCISR1,#%10000000,putchar   ; wait for transmit buffer empty
            staa  SCIDRL                      ; send a character
            rts
;***************end of putchar*****************

;*****************getchar************************
;* Program: Input one character from SCI port (terminal/keyboard)
;*             if a character is received, other wise return NULL
;* Input:   none
;* Output:  Accumulator A containing the received ASCII character
;*          if a character is received.
;*          Otherwise Accumulator A will contain a NULL character, $00.
;* Registers modified: CCR
;* Algorithm:
;    Check for receive buffer become full
;      Receive buffer full is indicated by RDRF bit
;      RDRF = 1 : full - Receive Data Register Full, 1 byte received
;      RDRF = 0 : not full, 0 byte received
;**********************************************
```

```
getchar     brclr SCISR1,#%00100000,getchar7
            ldaa  SCIDRL
            rts
getchar7    clra
            rts
;*****************end of getchar***************

;*****************nextline***********************
nextline
            psha
            ldaa  #CR              ; move the cursor to beginning of the line
            jsr   putchar          ;   Cariage Return/Enter key
            ldaa  #LF              ; move the cursor to next line, Line Feed
            jsr   putchar
            pula
            rts
;*****************end of nextline***************

;*****************delay1ms***********************
delay1ms:   pshx
            ldx   #$1000           ; count down X, $8FFF may be more than 10ms
d1msloop    nop                    ;   X <= X - 1
            dex                    ; simple loop
            bne   d1msloop
            pulx
            rts
;*****************end of delay1ms***************

msg3        DC.B   '> Be sure to start saving Terminal data: open Output file =
RxData3.txt', $00
msg4        DC.B   '> When ready, hit Enter/Return key for sawtooth wave, 1024 point
print.', $00
msg5        DC.B   '> Done!  You may close the Output file.', $00
msg6        DC.B   '> Ready for next data transmission, hit Enter key.', $00

            END                    ; this is end of assembly source file
                                   ; lines below are ignored - not assembled
```