

```

*equates-----
DATA      EQU $0000      ;start of data segment
MAIN      EQU $0100      ;start of program segment

*LCD display
COMBASE   EQU $B5F0
DATBASE   EQU $B5F1

*interrupts
TMSK1     EQU $1022
TFLG1     EQU $1023
TCNT      EQU $100E      ;free running counter

;output compare registers
TOC2      EQU $1018
TOC3      EQU $101A

;masks for manipulating output compare interrupts
OC2F      EQU %01000000
OC2I      EQU %01000000

OC3F      EQU %00100000
OC3I      EQU %00100000

;pseudo-vector location for OCs
OC2_VEC   EQU $00DC
OC3_VEC   EQU $00D9

*from HW4
DATA_LEN  EQU $0F
TWO       EQU $02
FOUR      EQU $04

TIME_BASE  EQU 2000      ;1ms
D_COUNT    EQU 4         ;4ms

*SCI system registers
SCCR2     EQU $102D      ;SCI Control Register 2
SCSR      EQU $102E      ;SCI Status Register
SCDR      EQU $102F      ;SCI Data Register
PORTA     EQU $1000

*data-----
      ORG DATA          ;initialize data
*LCD values from hw4
INIT_DATA  FCB $38        ;function set
          FCB $38        ;function set
          FCB $06        ;entry mode set
          FCB $01        ;clear display
          FCB $0F        ;display on/off control

CONT1      FCB $80        ; DD RAM Address Set to 000 0000
CONT2      FCB $C0        ; DD RAM Address Set to 100 0000

STRING     FCB 'i','n','i','t','i','a','l','#'

* Note the LCD data takes ASCII input directly
CNTR       RMB 1
counter    RMB 0
string_length RMB 0

*main initializations-----
      ORG MAIN
*JUMP TABLE INITIALIZATION
      LDAA    #$7E
      STAA    OC2_VEC
      LDD     #OC2_SVC
      STD     OC2_VEC+1

*TOC2 INITIALIZATION
      LDD     TCNT
      STD     TOC2

```

```

*OC2 INTERRUPT INITIALIZATION
    LDAA    #OC2I           ;enable OC2 in mak register
    STAA    TMSK1
    LDAA    #OC2F           ;clear OC3(2?) interrupt flag
    STAA    TFLG1
    CLI                      ;enable system interrupt

*LCD initialization
    LDAB    #FOUR
    LDX     #INIT_DATA

M1      LDAA    0,X           ;a=x
    STAA    COMBASE          ;display value x is pointing to

    LDAA    #D_COUNT         ;cntr=4
    STAA    CNTR
LOOP1   LDAA    CNTR          ;4ms delay
    BNE     LOOP1
    INX                     ;next block
    DECB
    BGE     M1               ;branch if greater than 0?
    ;BRA     BUTTON

*serial initalization
    ;LDAA    #$5             ;don't have to change baud rate yet...
    ;STAA    $102B           ;BRCR/BAUD
    CLR     $102C            ;SCCR1 (not used)
    CLR     SCDR             ;temp
    CLR     SCSR             ;temp
    LDAA    #%00000100       ;receive enable (RE), not sure if needed
    STAA    SCDR

*main code starts here-----
init    LDAA    #'0'
    STAA    counter
    LDAA    CONT1            ;init/clear first line of LCD
    STAA    COMBASE

button  LDAA    PORTA        ;wait until button is pushed
    CMPA    #$89
    BNE     button

*transmit string
    LDAA    #%000001000      ;transmit enable
    STAA    SCCR2

t_char  LDX     #string
    LDAA    0,X
    STAA    SCDR
    INX
    INC     string_length
    CMPA    #'#'
    JSR     trans
    BNE     t_char

*receive string
    LDAA    #%00000100       ;receive enable
    STAA    SCCR2

r_char  LDX     #string
    LDAA    SCDR
    INX
    ;STAA    DATABASE        ;temp
    STAA    0,X
    JSR     receive
    CMPA    #'#'
    CLR     SCDR             ;not sure if this is how receive works..
    BNE     r_char

*display
    LDX     #string           ;hopefully works all at once, if not incx and loop...cmp

```

```

to string_length too
    LDAA    0,X
    STAA    DATABASE

    LDAA    #FOUR           ;wait 4ms
    STAA    CNTR
wait4ms LDAA    CNTR
    BNE     wait4ms

    BRA     init

*subroutines-----
trans  LDAA    SCSR           ;wait until transmit is complete (TC)
    ANDA    #%01000000
    CMPA    #%01000000
    BEQ     return
    BRA     trans

receive LDAA    SCSR           ;wait until receive data register is full (RDRF)
    ANDA    #%00100000
    CMPA    #%00100000
    BEQ     return
    BRA     receive

return  RTS

*interrupts-----
OC2_SVC LDAA    #OC2F           ;SET UP TO
    STAA    TFLG1           ;CLEAR OC2F
    LDD     TOC2           ;SET UP FOR NEXT INTERRUPT
    ADDD    #TIME_BASE      ;ADD TIMEBASE (1 MS)
    STD     TOC2           ;AND STORE

*-----
OC2RTI  DEC     CNTR           ;DEC TIME BASE COUNTER
    RTI                    ;EXIT

```

