

```

list
; E16F84.INC Enhanced Header File, Version 1.00 Eagle Air Australia P/L
nolist

;*****
; E16C84.INC Enhanced Header File, Version 1.00 *
; This header file provides enhanced defines for PIC 16C84 assembler *
; It replaces p16f84.inc *
; *
; Author : John F. Fitter B.E. *
; Revisions : 17jul96 Original *
; 23jul96 Consolidated p16f84 defines and added new macros *
; *
; Copyright c 1996 Eagle Air Australia Pty. Ltd. All rights reserved *
; (some code fragments are based on application notes and other public sources) *
;*****

        ifndef    _E16F84
        #define   _E16F84

        ifndef    __16F84
        MESSG     "Processor-header file mismatch. Verify selected processor."
        endif

true     equ      1
false   equ      0

input   equ      1
output  equ      0

_resvec equ      0x00      ; 16f84 start address
_intvec equ      0x04      ; 16f84 peripheral interrupt vector
_endpgm equ      0x3FF     ; end of program memory

resetvector org    _resvec      ; reset vector
resvec      goto   init

endmemory   org    _endpgm      ; end of memory
endpgm

;*****
; Macros to implement new addressing modes *
;*****

w        equ      0          ; Working register destination
W        equ      w
f        equ      1          ; File register destination
F        equ      f

movff    macro      src, dest      ; Move register file src to register
        movf        src, W          ; file dest through w
        movwf       (dest)&0x7f
        endm

movlf    macro      dest, k        ; Move literal to register file
        movlw       k
        movwf       (dest)&0x7f
        endm

andlf    macro      dest, k        ; AND literal with register file
        movlw       k
        andwf       (dest)&0x7f, f
        endm

iorlf    macro      dest, k        ; Inclusive OR literal with register file
        movlw       k
        iorwf       (dest)&0x7f, f
        endm

xorlf    macro      dest, k        ; Exclusive OR literal with register file
        movlw       k
        xorwf       (dest)&0x7f, f
        endm

addlf    macro      dest, k        ; ADD literal to register file
        movlw       k
        addwf       (dest)&0x7f, f
        endm

sublf    macro      dest, k        ; SUBTRACT literal from register file
        movlw       k
        subwf       (dest)&0x7f, f
        endm

```

```

;*****
; Macros to set/clear/branch/skip on bits
; These macros define and use synthetic "bit labels"
; Bit labels contain the address and bit of a location
;*****
;
; Usage Description
; -----
; bit label, bit, file ; Define a bit label
; seb label ; set bit using bit label
; clb label ; clear bit using bit label
; skbs label ; SKIP on bit set
; skbc label ; SKIP on bit clear
; bbs label, address ; BRANCH on bit set
; bbc label, address ; BRANCH on bit clear
; cbs label, address ; CALL on bit set
; cbc label, address ; CALL on bit clear

bit macro label, bit, file ; Define a bit label
label equ ((file)&0x7f)<<8|bit; (macro)
endm

seb macro label ; Set bit
bsf label>>8, label&7 ; (macro)
endm

clb macro label ; Clear bit
bcf label>>8, label&7 ; (macro)
endm

skbs macro label ; Skip on bit set
btfss label>>8, label&7 ; (macro)
endm

skbc macro label ; Skip on bit clear
btfsc label>>8, label&7 ; (macro)
endm

bbs macro label, address ; Branch on bit set
btfsc label>>8, label&7 ; (macro)
goto address ; (macro)
endm

bbc macro label, address ; Branch on bit clear
btfss label>>8, label&7 ; (macro)
goto address ; (macro)
endm

cbs macro label, address ; Call on bit set
btfsc label>>8, label&7 ; (macro)
call address ; (macro)
endm

cbc macro label, address ; Call on bit clear
btfss label>>8, label&7 ; (macro)
call address ; (macro)
endm

;*****
; REGISTER FILE DECLARATIONS
;*****

INDF equ 0x00 ; Uses contents of FSR to address data memory
TMRO equ 0x01 ; Timer 0
PCL equ 0x02 ; Program counter low order 8 bits
STATUS equ 0x03 ; Status register
FSR equ 0x04 ; Indirect data memory address pointer
PORTA equ 0x05 ; Port A
PORTB equ 0x06 ; Port B
EEDATA equ 0x08 ; EEPROM data register
EEADR equ 0x09 ; EEPROM address register
PCLATH equ 0x0A ; Write buffer for upper 5 bits of PC
INTCON equ 0x0B ; Interrupt control register

OPTION_REG equ 0x81 ; Option register
TRISA equ 0x85 ; Port A data direction register
TRISB equ 0x86 ; Port B data direction register
EECON1 equ 0x88 ; EEPROM control register 1
EECON2 equ 0x89 ; EEPROM control register 2

CONFIG_REG equ 0x2007 ; Configuration word (no access by program)

;*****
; REGISTER BIT DECLARATIONS
;*****

```

```

bit    B_C,      0, STATUS      ; Carry
bit    B_DC,    1, STATUS      ; Half carry
bit    B_Z,     2, STATUS      ; Zero
bit    B_PD,    3, STATUS      ; Power down
bit    B_TO,    4, STATUS      ; Timeout
bit    B_RP0,   5, STATUS      ; Register bank select bit 0 (direct)
bit    B_RP1,   6, STATUS      ; Register bank select bit 1 (direct)
bit    B_IRP,   7, STATUS      ; Register bank select bit (indirect)

bit    B_RBIF,  0, INTCON      ; RB port change interrupt flag
bit    B_INTF,  1, INTCON      ; RB0/INT interrupt flag
bit    B_TOIF,  2, INTCON      ; TMR0 overflow interrupt flag
bit    B_RBIE,  3, INTCON      ; RB port change interrupt enable
bit    B_INTE,  4, INTCON      ; RB0/INT interrupt enable
bit    B_TOIE,  5, INTCON      ; TMR0 overflow interrupt enable
bit    B_EEIE,  6, INTCON      ; EE write complete interrupt enable
bit    B_GIE,   7, INTCON      ; Global interrupt enable

bit    B_PS0,   0, OPTION_REG  ; prescaler bit 0
bit    B_PS1,   1, OPTION_REG  ; prescaler bit 1
bit    B_PS2,   2, OPTION_REG  ; prescaler bit 2
bit    B_PSA,   3, OPTION_REG  ; prescaler assignment
bit    B_T0SE,  4, OPTION_REG  ; TMR0 source edge select
bit    B_T0CS,  5, OPTION_REG  ; TMR0 clock source select
bit    B_INTEDG,6, OPTION_REG  ; Interrupt edge select
bit    B_NOT_RBPU,7, OPTION_REG ; Port B pullup enable

bit    B_RD,    0, EECON1      ; EEPROM Read control bit
bit    B_WR,    1, EECON1      ; EEPROM Write control bit
bit    B_WREN,  2, EECON1      ; EEPROM Write enable bit
bit    B_WRERR, 3, EECON1      ; EEPROM Error flag bit
bit    B_EEIF,  4, EECON1      ; EEPROM Write operation interrupt flag

;*****
; RAM Definition
;*****

    __MAXRAM    H'CF'
    __BADRAM    H'07', H'50'-H'7F', H'87'

;*****
; Configuration Bits
;*****

_CP_ON      EQU    H'000F'
_CP_OFF     EQU    H'3FFF'
_PWRTE_ON   EQU    H'3FF7'
_PWRTE_OFF  EQU    H'3FFF'
_WDT_ON     EQU    H'3FFF'
_WDT_OFF    EQU    H'3FFB'
_LP_OSC     EQU    H'3FFC'
_XT_OSC     EQU    H'3FFD'
_HS_OSC     EQU    H'3FFE'
_RC_OSC     EQU    H'3FFF'

endif ; _E16F84
list

```