

```

;*****
; This file is a basic code template for object module code      *
; generation on the PICmicro PIC10F206. This file contains the  *
; basic code building blocks to build upon. As a project minimum *
; the 10F206.lkr file will also be required for this file to    *
; correctly build. The .lkr files are located in the            *
; MPLAB/MCHIP_Tools directory.                                  *
;                                                                *
; If the internal RC oscillator is not implemented then the first *
; instruction after the code section "MAIN" is not required.    *
;                                                                *
; Refer to the MPASM User's Guide for additional information on  *
; features of the assembler and linker (Document DS33014F).    *
;                                                                *
; Refer to the respective PICmicro data sheet for additional    *
; information on the instruction set.                            *
;                                                                *
; Template file built using MPLAB V3.99.18 with MPASM V2.15.06 and *
; MPLINK 1.10.11 as the language tools.                        *
;*****
;
; Filename:      blinker.asm                                     *
; Date:         9/8/2004                                       *
; File Version: 1.00.00                                         *
;
; Author:       Anthony Voss                                    *
; Company:      Newark In One                                   *
;
;*****
;
; Files required:
;                MPLAB/MCHIP_Tools/10f206.lkr
;
;*****
;
; Notes:
;                Blinks led D6 on pickit 1
;
;
;*****

list      p=10f206          ; list directive to define processor
#include <p10f206.inc>      ; processor specific variable definitions

__CONFIG  _CP_OFF & _WDT_OFF & _MCLRE_ON & _IntRC_OSC

; '__CONFIG' directive is used to embed configuration word within .asm file.
; The labels following the directive are located in the respective .inc file.
; See respective data sheet for additional information on configuration word.

;***** VARIABLE DEFINITIONS
;TEMP_VAR      UDATA      0x10
temp           EQU        0x10          ;example variable definition

```

```

delay1          EQU          0x11          ;delay counter
delay2          EQU          0x12          ;delay counter

```

```

;*****
                org          0x3FF
;                CODE      0x3FF          ; processor reset vector
; Internal RC calibration value is placed at location 0x3FF by Microchip
; as a movlw k, where the k is a literal value.

                org          0x00
movwf          OSCCAL          ;load calibration value into
bcf            OSCCAL,FOSC4    ;disable fosc/4 output on GP
goto          MAIN

; =====
;subroutines
delay
                movlw       0xfa          ;load delay1 counter
                movwf      delay1
                movlw       0xFF         ;load delay2 counter
                movwf      delay2

delay_loop1
                decfsz     delay1,F      ;decrement delay1 until zero
                goto       delay_loop2   ;goto nested inner loop
                retlw      0x00         ;return from the call

delay_loop2
                decfsz     delay2,F      ;decrement delay2 until zero
                goto       delay_loop2   ;not done decrementing. Stay
                goto       delay_loop1   ;done decrementing. Return t

; =====

MAIN                ; coding begins here
                movlw      b'11000000'   ;set option to no wake-up, n
                option
                movlw      b'11110011'   ;set comparator to output no
                movwf      CMCON0
                movlw      b'00000000'   ;clear the data latch
                movwf      GPIO
                movlw      b'11111001'   ;set GP <2:1> to be write. A
                tris       GPIO

Loop_here
                movlw      b'00000100'   ;bit toggle LED by xor-ing w
                xorwf      GPIO,F
                call       delay
                goto       Loop_here

                END                ; directive 'end of program'

```