;====================================================================

;

; Author : ADI - Apps

;

; Date : January 2001

;

; File : Tim2DeB.asm

;

; Hardware : ADuC816

;

; Description : Code example to show how the debugger can be 'halted'.

; i.e. if user code does not appear to be executing

; correctly and seems to have missed the breakpoints

; then you can halt (stop) the ADuC816 from running

; and examine where code is being executed from.

;

; The debugger uses timer 2 as the baudrate generater

; on the ADuC816. When timer 2 is used as a baudrate

; generator it does not generate timer 2 interrupts

; using the TF2 flag. However the EXF2 flag can still

; be used to generate a 3rd external interrupt. Usually

; this extra option is not used however.

;

; In order to halt the ADuC816 the following code must

; be included in code before it is assembled. A

; breakpoint must be set on the RETI instruction of

; the TIMER 2 ISR. Hence if the code goes 'missing'

; and the T2EX pin (Pin 2) grounded then a Timer 2

; interrupt will occur hitting your breakpoint. Single

; stepping over the RETI instruction will show you

; where the code was executing.

;

; NOTE: On the ADuC816 evalutaion board the T2EX pin

; appears at J3.7. A ground connection is provided

; at J2.8 directly across from it. Hence connecting

; a link here will cause the timer 2 interrupt.

;

; The following code example shows how this approach

; could be used around the simple blink routine

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

$MOD816 ;Use 8052 predefined Symbols

LED EQU P3.4

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; BEGINNING OF CODE

CSEG

ORG 0000H

JMP MAIN

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; INT0 ISR

ORG 0003h

INC A ; Increment Acc

RETI ; Return from Interrupt

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; TIMER 2 INTERRUPT ROUTINE

ORG 002Bh

CLR EXF2

RETI ; <-- SET B'POINt here in Debugger

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; MAIN PROGRAM

MAIN:

; the following 3 lines configures the timer 2 interrupt to halt the

; ADuC816. Note if the SETB EA instruction is already in the code

; then this is unnecessary

ORL T2CON, #08h ; set EXEN2 without effecting baud rate

SETB ET2

SETB EA

; USER CODE GOES HERE......

SETB IT0 ; INT0 edge triggered

SETB EA ; enable inturrupts

SETB EX0 ; enable INT0

MOV A,#01H ; Initialize A -> 1

BLINK:

CPL LED ; blink LED using compliment instruction

CALL DELAY ; Call subroutine DELAY

JMP BLINK

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; DELAY

DELAY: ; Delays by 100ms \* A

MOV R0,A ; Acc holds delay variable

DLY0: MOV R1,#019h ; Set up delay loop0

DLY1: MOV R2,#0FEh ; Set up delay loop1

DJNZ R2,$ ; Dec R2 & Jump here until R2 is 0

DJNZ R1,DLY1 ; Dec R1 & Jump DLY1 until R1 is 0

DJNZ R0,DLY0 ; Dec R0 & Jump DLY0 until R0 is 0

RET ; Return from subroutine

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

END