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

;

; Author : ADI - Apps

;

; Date : October 2003

;

; File : adctrig.asm

;

; Description : Flash led an initial rate of 100ms

; Pressing INTO triggers single conversion

; The ADC result is written to internal memory

; The delay rate is increased

; The program waits for the next INTO to repeat the

; above sequence

;

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

;

$MOD842 ; Use ADuC842 predefined Symbols

FLAG EQU 00H ; Define Bit

CHAN EQU 00H

DSEG

ORG 0030H

LENGTH EQU 40

BUFFER: DS LENGTH

CSEG ; Defines the following as a segment of code

ORG 0000H ; Load Code at '0'

JMP MAIN ; Jump to MAIN

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

ORG 0003h ; (INT0 ISR)

MOV B,A ; Copy A (sets delay)

INC A ; Increment delay

SETB SCONV ; INITIATE A MAIN ADC SINGLE CONVERSION

JNB ADCI,$ ; Wait for conversion results

; Write ADC Result to memory

MOV @R0,ADCDATAH

INC R0

MOV @R0,ADCDATAL

INC R0

MOV A,B ; Restore A (sets delay)

INC A ; Increment delay

RETI ; Return from Interrupt

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

ORG 004Bh ; Subroutines

;------------------------------------------------------------------

DELAY: ; Delays by 10ms \* A

; 10mSec based on 2.09MHZ

; Core Clock

; i.e. default ADuC842 Clock

MOV R1,A ; Acc holds delay variable (1 clock)

DLY0: MOV R2,#01Bh ; Set up delay loop0 (2 clocks)

DLY1: MOV R3,#0FFh ; Set up delay loop1 (2 clocks)

DJNZ R3,$ ; Dec R3 & Jump here until R3 is 0 (3 clocks)

DJNZ R2,DLY1 ; Dec R2 & Jump DLY1 until R2 is 0 (3 clocks)

DJNZ R1,DLY0 ; Dec R1 & Jump DLY0 until R1 is 0 (3 clocks)

RET ; Return from subroutine

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

MAIN: ; (main program)

; Configure ADC

MOV ADCCON1,#0ACh ; power up ADC

MOV ADCCON2,#CHAN ; select channel to convert

MOV R0,#BUFFER

SETB IT0 ; INT0 edge triggered

SETB EA ; enable inturrupts

SETB EX0 ; enable INT0

MOV A,#010 ; Initialize A -> 10

BLINK: CPL P3.4 ; blink LED using compliment instruction

CALL DELAY ; Jump to subroutine DELAY

JMP BLINK ; If FLAG is still cleared the jump to Blink

END