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

;

; Author : ADI - Apps

;

; Date : 5 November 2001

;

; File : adcconv.asm

;

; Hardware : ADuC836

;

; Description : example routine to continuosuly trigger

; a single conversion on the ADC main channel

; The Conversion result is written to external memory,

; P3.4 is toggled 5 times at 100ms delay

; The ADC result is written to external memory

; P3.4 is toggled 5 times at 500ms

; and the sequence repeats itself.

;

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

;

$MOD836 ; Use 8052&ADuC836 predefined symbols

FLAG EQU 00H ; Define Bit

CSEG ; Defines the following as a segment of code

ORG 0000H ; Load Code at '0'

JMP START ; Jump to START

;

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

;

ORG 0033H ; Read ADC Result H/M/L to external Memory

; 10 flashes at 100ms

MOV A,#01H ; 100msec delay

MOV R3,#0Ah ; loop=5

TIC1: CPL P3.4 ; Toggle LED

CALL DELAY ; Delay 100mSec

DJNZ R3,TIC1 ; Dec loop

MOV DPTR, #00H ; DPTR=00

MOV A,ADC0M ; read ADC Middle byte

MOVX @DPTR,A ; write Middle byte to ext memory

INC DPTR ; DPTR=01

MOV A,ADC0H ; read ADC High byte

MOVX @DPTR,A ; write low High byte to ext memory

INC DPTR

; 5 flashes at 500ms

MOV A,#05H ; 500msec delay

MOV R3,#0Ah ; loop=10

TIC2: CPL P3.4 ; Toggle LED

CALL Delay ; Delay 100mSec

DJNZ R3,TIC2 ; Dec loop

;

SETB Flag ; Set Flag

RETI ; Return from Interrupt

;

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

;

ORG 0060H

START:

; Configure Interrupt System

SETB IT0 ; CONFIG EXTERNAL INTERRUPT Falling Edge

SETB EA ; Enable Global Interrupts

; Configure ADC

MOV ADCMODE,#20H ; ENABLE MAIN ADC; Mode- Power down

MOV ADC0CON,#40H ; 24 BITS

; USE EXTERNAL REFERENCE

; AIN1-AIN2

; BIPOLAR MODE

; RANGE = +/-20mV

SETB EADC ; ENABLE ADC INTERRUPT

; Looped single conversions

CONV: CLR FLAG ; Initial condition for FLAG variable

MOV ADCMODE,#22H ; INITIATE A MAIN ADC SINGLE CONVERSION

CALL DELAY ; Jump to subroutine DELAY

JNB FLAG,$ ; Stay here until FLAG=1 i.e. wait for ADC Int.

JMP CONV ; Next ADC Conversion

;

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

;

DELAY: ; Delays by A\*100ms (default Core Clk =1.57MHz)

MOV R0,A ; Acc holds delay variable

DLY0: MOV R1,#0FEh ; Set up delay loop0

DLY1: MOV R2,#019h ; 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