;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; Author : ADI - Apps www.analog.com/MicroConverter

;

; Date : October 2003

;

; File : ADCcont.asm

;

; Hardware : ADuC842/ADuC843

;

; Description : Performs ADC conversions in continuous mode at a

; rate of 104KSPS.

; Outputs ADC results into a buffer in ram.Continuously

; flashes LED.

; All rate calculations assume an 2.097152MHz Mclk.

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$MOD842 ; Use 8052&ADuC842 predefined symbols

LED EQU P3.4 ; P3.4 drives red LED on eval board

CHAN EQU 0 ; convert this ADC input channel..

; ..chan values can be 0 thru 8

DSEG

ORG 0030H

LENGTH EQU 40

BUFFER: DS LENGTH ; set up buffer in RAM

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

; BEGINNING OF CODE

CSEG

ORG 0000h

JMP MAIN ; jump to main program

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

; INTERRUPT VECTOR SPACE

ORG 0033H ; (ADC ISR)

CJNE R0,#58H,CONT

JMP EXIT ; place breakpoint here in debugger to view ram after conversions

CONT: MOV @R0,ADCDATAH

INC R0

MOV @R0,ADCDATAL

INC R0

EXIT: RETI

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

; MAIN PROGRAM

ORG 004Bh

MAIN:

; PRECONFIGURE...

MOV ADCCON1,#0ACh ; power up ADC /8 clk cycle + 16 cycles for conversion + 4 acq cycle

MOV ADCCON2,#CHAN ; select channel to convert

; LAUNCH CONTINUOUS CONVERSIONS...

MOV R0,#BUFFER

SETB EA ; enable interrupts

SETB EADC ; enable ADC interrupt

SETB CCONV ; begin continuous conversions

; CONTINUE WITH OTHER CODE...

MOV A,#10 ; delay length

AGAIN: CPL LED ; blink (complement) the LED

CALL DELAY ; delay

JMP AGAIN ; repeat

; the micro is free to continue with other tasks (flashing the LED in

; this case) while the ADC is continuously converting, and results

; are being handled by the ADC interrupt service routine.

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

; SUBROUTINE

DELAY: ; Delays by 10ms \* A

; 10mSec based on 2.09MHz

; Core 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

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

END