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

;

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

;

; Date : 5 November 2001

;

; File : 836uart2.asm

;

; Hardware : ADuC836

;

; Description : sample program that performs ADC conversions in

; continuous mode and sends results to a PC via the

; UART. program also accepts commands from the PC

; via incoming characters on the UART. an ASCII "0"

; innitiates a zero-scale calibration, and an ASCII

; "1" innitiates a full-scale calibration.

;

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

$MOD836 ; Use 8052&ADuC836 predefined symbols

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

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

; BEGINNING OF CODE

CSEG

ORG 0000h

JMP MAIN

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

; INTERRUPT VECTOR SPACE

ORG 0033h ; (ADC ISR)

CPL LED

MOV DPTR,#SEPERATOR ; send linefeed+CR out UART

CALL SENDSTRING

MOV A,ADC0H ; send ADC data via UART

CALL SENDVAL

MOV A,ADC0M

CALL SENDVAL

CLR RDY0

RETI

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

; MAIN PROGRAM

ORG 0100h

MAIN:

MOV SP,#127

; CONFIGURE UART....

MOV T3CON,#82h

MOV T3FD,#12h

MOV SCON,#52h

; CONFIGURE ADC....

MOV SF,#200 ; 6.8266667Hz ADC data rate

MOV ADC0CON,#045h ; externalVref, bipolar, ±640mV

SETB EADC ; enable ADC interrupt (trig on RDY0)

; START CONVERTING & WAIT FOR INTERRUPTS OR INCOMING UART COMS....

WAIT: SETB EA

CLR RDY0

MOV ADCMODE,#023h ; continuous conversion mode

CLR RI

JNB RI,$ ; wait here for UART command..

; ..or ADC interrupt

; WHEN UART COMMAND RECEIVED....

CLR EA ; disable interrupts

MOV ADCMODE,#021h ; put ADC in idle mode

MOV A,SBUF

CJNE A,#'0',NEXT01 ; if "0" received..

CALL CALZERO ; ..perform zero calibration

JMP WAIT

NEXT01: CJNE A,#'1',WAIT ; if "1" received..

CALL CALGAIN ; ..perform gain calibration

JMP WAIT

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

; SUBROUTINES

CALZERO: ; perform zero-scale ADC calibration

MOV DPTR,#CALZSMSG ; send char to indicate..

CALL SENDSTRING ; ..begin zero calibration

CLR RDY0

MOV ADCMODE,#034 ; zero-scale self cal

; MOV ADCMODE,#036h ; zero-scale system cal

JNB RDY0,$

CLR RDY0

MOV DPTR,#CALDONEMSG ; send char to indicate..

CALL SENDSTRING ; ..calibraion complete

RET

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

CALGAIN: ; perform full-scale ADC system calibration

MOV DPTR,#CALGNMSG ; send char to indicate..

CALL SENDSTRING ; ..begin gain calibration

CLR RDY0

MOV ADCMODE,#035 ; full-scale self cal

; MOV ADCMODE,#037h ; full-scale system cal

JNB RDY0,$

CLR RDY0

MOV DPTR,#CALDONEMSG ; send char to indicate..

CALL SENDSTRING ; ..calibration complete

RET

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

; SUBROUTINE INCLUDE FILE

$INCLUDE(UARTIO.asm)

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

; TEXT DATA TABLES

SEPERATOR: DB 10,13,0

CALZSMSG: DB 'Z',0

CALGNMSG: DB 'G',0

CALDONEMSG: DB 'x',0

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

END