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

;

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

;

; Date : October 2003

;

; File : dataflsh.asm

;

; Hardware : ADuC842/ADuC843

;

; Description : Demonstrates use of the on-chip read/write 4096 byte

; FlashEE data memory space. Stores a sequence of

; button presses (INT0 button on eval board) in data

; FlashEE space. Replays sequence on LED when board

; is reset or power cycled.

; The ADuC842 stores the play sequece in data flash

; until another is recorded with a new set of button

; presses. To record a new sequence, just wait until

; the current one finishes playing (LED is off) and

; enter new sequence via button (INT0).

;

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

$MOD842 ; Use 8052&ADuC842 predefined symbols

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

BUTTON EQU P3.2 ; button on eval board drives P3.2

PREVIOUS EQU F0 ; flag to hold previous button value

READ EQU 01h ; FlashEE command: 'read page'

WRITE EQU 02h ; FlashEE command: 'write page'

VERIFY EQU 04h ; FlashEE command: 'verify page'

ERASE EQU 05h ; FlashEE command: 'erase page'

ERASEALL EQU 06h ; FlashEE command: 'erase all'

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

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

; BEGINNING OF CODE

CSEG

ORG 0000h

MAIN:

CLR LED ; turn LED off

MOV EADRH,#0 ; set data FlashEE address to page 0

MOV EADRL,#0

; READ FLASH/EE DATA and indicate values via LED on and off times...

READFLASH:

MOV ECON,#READ ; read current 4byte page of FlashEE

; into EDATA1,2,3,4

MOV A,EDATA4

CJNE A,#1,RECORD ; if EDATA4 is 1, then page contains

; a valid play sequence

; => Play this sequence

; otherwise jump to record mode

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

PLAYBACK:

CALL BLINK ; flash LED for period determined

; by FlashEE data just read

MOV A,EADRL

CJNE A,#0FFh,INCPAGE1 ; if low address is FFh then increment high address

INC EADRH

INCPAGE1:

INC EADRL ; increment to next FlashEE page addr

MOV A,EADRH

CJNE A,#04h,READFLASH

; if address is less than 160 then jump

; to read the next page

; when PLAYBACK is finished jump to RECORD mode

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

RECORD:

CLR LED

JB BUTTON,$ ; wait for first button press

; once button is pressed, erase dataflash

MOV ECON,#ERASEALL ; clear all data FlashEE memory

MOV EADRH,#0

MOV EADRL,#0

; record time of button press

RECORD\_NEXT\_TIME:

CALL RECORDTIME

MOV EDATA1,DPL ; place DPTR in EDATA1,2,3

MOV EDATA2,DPH

MOV EDATA3,DPP

MOV EDATA4,#1 ; put 1 in EDATA4 as identifier

MOV ECON,#WRITE ; write EDATA1-4 into pre-erased

; page of FlashEE data memory

MOV ECON,#VERIFY ; verify current page is same as

MOV A,ECON ; EDATA1-4. If same, ECON=0.

JNZ RECORD ; if verify fails, jump to RECORD

MOV A,EADRL

CJNE A,#0FFh,INCPAGE2 ; if low address is FFh then increment high address

INC EADRH

INCPAGE2:

INC EADRL ; increment to next FlashEE page addr

MOV A,EADRH

CJNE A,#04h,RECORD\_NEXT\_TIME

; record first 160 button presses only

; when flash/EE data space is full turn off LED and wait

; for a power cycle

CLR LED

JMP $

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

; FUNCTIONS

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

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

; SUBROUTINES

BLINK:

; Turn LED ON/OFF based on the time in EDATA3/2/1

CPL LED

CLR A

MOV DPL,A

MOV DPH,A ; clear DPTR

MOV DPP,A

INC EDATA1 ; EDATA1 -> EDATA3 should be

INC EDATA2 ; incremented for the below to work

INC EDATA3

BLINKLOOP:

; the record loop takes 11 instruction cycles hence 7 NOPs are

; required to make the Playback loop 11 instruction cycles also.

; NOTE: the main Playback loop will jump to BLINKLOOP after

; decrementing EDATA1 and hence the time required to decrement

; EDATA2 (approx 1/256 time of main loop) and EDATA3 are ignored.

NOP ; 1

NOP ; 1

NOP ; 1

NOP ; 1

NOP ; 1

NOP ; 1

NOP ; 1

DJNZ EDATA1, BLINKLOOP ; 4

DJNZ EDATA2, BLINKLOOP ; EDATA1 overflows back to FFh

DJNZ EDATA3, BLINKLOOP ; EDATA2 overflows back to FFh

RET

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

RECORDTIME:

; Record how long button is pressed for and store in EDATA3/2/1

CLR A

MOV DPL,A

MOV DPH,A ; clear DPTR

MOV DPP,A

CPL LED

; measure how long the button is either pressed or released

; for. If the button is pressed then the LED is on. If the

; button is released then the LED is off.

RECORDLOOP:

INC DPTR ; incrementing DPTR.. ; 3

JNB LED, CHKB ; 4

JNB BUTTON,RECORDLOOP ; 4

; keep recording while button is pressed

RET

CHKB: JB BUTTON,RECORDLOOP ; 4

; keep recording while button is released

RET

; DPP,DPH,DPL now holds a number that represents the length of

; time between button edges. this data will be stored in FlashEE

; space for use in controlling LED on and off times in "play" mode.

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

END