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

;

; Author : ADI - Video Apps

;

; Date : 12/07/99

;

; File : tempLCD.asm

;

; Description : This program is outputs the temperature from the OnChip

; Temp Sensor.

;

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

$INCLUDE(MOD824) ;Use 8052 predefined Symbols

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

;

; Definitions

;

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

LCD\_RS EQU P2.6

LCD\_WR EQU P2.7

LCD\_EN EQU P3.6

LCD\_DATA EQU P0

CntrByte1 EQU 08Eh

CHAR DATA 040H ; Char buffer for LCD

D4100h EQU 01Ch ; Timer reload high byte def.

D4100l EQU 013h ; Timer reload = 4.1ms

ERASE EQU 015h

READ EQU 011h

FWRITE EQU 012h

DSEG

ORG 0033h

DIG1: DS 1

DIG2: DS 1

DIG3: DS 1

DIGC1: DS 1

DIGC2: DS 1

DIGC3: DS 1

NEG: DS 1

TEMP: DS 1

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

;

; Reset and interrupt vectors

;

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

CSEG

DISPLAY: MOV SP,#070h

MOV R3,#18h

; Display initialization

MOV R0,#038H ; Functiun set

CALL SEND\_CMD1 ;

MOV R0,#06H ; Entry set

CALL SEND\_CMD1 ;

MOV R0,#02H ; Home

CALL SEND\_CMD1 ;

MOV R0,#0CH ; Display on, cursor, etc

CALL SEND\_CMD

;MOV ADC1CON,#20H ; Temperature sensor initialisation

MOV R1,#00h

MOV DPTR,#TEXT0 ; Welcome message stored in code memory

LCALL BOTHLINES

LCALL WAIT ; Delay: 2 secondes

MOV R1,#00h

MOV DPTR,#TEXT1 ; Temp monitor message stored in code memory

LCALL BOTHLINES

LCALL WAIT

MOV R1,#00h

MOV DPTR,#TEXT2 ; Temperature message stored in code memory

LCALL BOTHLINES

AGAIN: ;MOV ADCMODE,#12H ; Initiate a single aux conversion

;JNB RDY1,$ ; Wait for conversion results

;CLR RDY1

LCALL TEMPER

MOV R0,#88H ; Address on the display

CALL SEND\_CMD ; Send address to LCD

MOV R0,DIGC3 ; Number to display in R0

CALL SEND\_CHAR ; Display DIGC3

MOV R0,#89H ; Address on the display

CALL SEND\_CMD ; Send address to LCD

MOV R0,DIGC2 ; Number to display in R0

CALL SEND\_CHAR ; Display DIGC2

MOV R0,#8AH ; Address on the display

CALL SEND\_CMD ; Send address to LCD

MOV R0,DIGC1 ; Number to display in R0

CALL SEND\_CHAR ; Display DIGC1

MOV R0,#0C8H ; Address on the display

CALL SEND\_CMD ; Send address to LCD

MOV R0,DIG3 ; Number to display in R0

CALL SEND\_CHAR ; Display DIG3

MOV R0,#0C9H ; Address on the display

CALL SEND\_CMD ; Send address to LCD

MOV R0,DIG2 ; Number to display in R0

CALL SEND\_CHAR ; Display DIG2

MOV R0,#0CAH ; Address on the display

CALL SEND\_CMD ; Send address to LCD

MOV R0,DIG1 ; Number to display in R0

CALL SEND\_CHAR ; Display DIG1

MOV R0,#87H

LCALL CLRCHAR

MOV R0,#0C7H

LCALL CLRCHAR

MOV R0,NEG

CJNE R0,#00H,INF0 ; If NEG<>0 then MINUSC

JMP FINISH

INF0: LCALL MINUSC

MOV R0,NEG

CJNE R0,#12H,INF01 ; If NEG>17 then MINUSF

INF01: JNC INF02

JMP FINISH

INF02: LCALL MINUSF

FINISH: JMP AGAIN

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

;

; Subroutines for the LCD display

;

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

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

;

; SEND\_CMD: SENDS COMMANDS TO LCD

;

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

SEND\_CMD: MOV CHAR,R0 ; SAVES R0 TO LOCATION CHAR

CALL CHKBUSY ; CHECKS BUSY FLAG

MOV R0,CHAR ; PULLS CONTENTS OF R0 FROM CHAR

MOV LCD\_DATA,R0 ; SENDS TO PORT 0

CLR LCD\_WR ; CLEARS R/W LINE

CLR LCD\_RS ; CLEARS RS LINE

CALL PULSE\_E ; CLOCKS IN DATA INTO LCD

RET ; RETURNS

SEND\_CMD1: MOV CHAR,R0 ; SAVES R0 TO LOCATION CHAR

CALL LDELAY ; CHECKS BUSY FLAG

MOV R0,CHAR ; PULLS CONTENTS OF R0 FROM CHAR

MOV LCD\_DATA,R0 ; SENDS TO PORT 0

CLR LCD\_WR ; CLEARS R/W LINE

CLR LCD\_RS ; CLEARS RS LINE

CALL PULSE\_E ; CLOCKS IN DATA INTO LCD

RET ; RETURNS

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

;

; SEND\_CHAR: SENDS CHARACTERS TO LCD

;

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

SEND\_CHAR: MOV CHAR,R0 ; SAVES R0 TO LOCATION CHAR

CALL CHKBUSY ; CHECKS BUSY FLAG

MOV R0,CHAR ; PULLS CONTENTS OF R0 FROM CHAR

MOV LCD\_DATA,R0 ; SENDS TO PORT 0

CLR LCD\_WR ; CLEARS R/W LINE

SETB LCD\_RS ; SETS RS LINE HIGH FOR DATA

CALL PULSE\_E ; CLOCKS DATA INTO LCD

RET

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

;

; CHKBUSY: CHECKS BUSY FLAG OF LCD

;

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

CHKBUSY: MOV R0,#0FFH

MOV LCD\_DATA,R0 ; SETS UP PORT0 AS I/P'S

CLR LCD\_RS ; CLEARS RS LINE

SETB LCD\_WR ; SETS R/W LINE

NOP

NOP

NOP

BUSYREAD: SETB LCD\_EN ; CLOCKS E HIGH

NOP

NOP

NOP

MOV ACC,LCD\_DATA ; READS PORT 2

NOP

NOP

NOP

CLR LCD\_EN ; CLOCKS E LOW

NOP

NOP

JB ACC.7,BUSYREAD ; IF BUSY, LOOP

RET

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

;

; PULSE\_E: GENERATES CLOCK SIGNAL, E.

;

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

PULSE\_E: NOP

NOP

SETB LCD\_EN ; CLOCKS E HIGH

NOP

NOP

NOP

NOP

NOP

CLR LCD\_EN ; CLOCKS E LOW

NOP

NOP

RET

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

;

; Subroutine: WRITE

;

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

WRITE:

write\_loop: MOV A,R1

MOVC A,@A+DPTR

CJNE A,#99h,write\_cont ; stop writing condition

RET

write\_cont: MOV R0,A

CALL send\_char ; send one ASCII character

INC DPTR ; next character

JMP write\_loop

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

;

; Data bytes

;

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

TEXT0: DB ' ANALOG DEVICES ' ; welcome message

DB 099h

DB ' MICROCONVERTER '

DB 099h

TEXT1: DB ' ADuC824 ' ; temp monitor message

DB 099h

DB ' TEMP MONITOR '

DB 099h

TEXT2: DB ' TEMP = ' ; temperature message

DB 0DFH

DB 'C '

DB 099h

DB ' TEMP = '

DB 0DFH

DB 'F '

DB 099h

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

;

; WRITE IN THE EEFLASH

;

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

EEFLASH: MOV EADRL,#00h ; PAGE 0

MOV EDATA1,R1 ; Store number of channel

MOV EDATA2,R5 ; R5=DPL

MOV EDATA3,R6 ; R6=DPH

MOV EDATA4,#01h

MOV ECON,#ERASE ; Erase page 0 before writing

MOV ECON,#FWRITE

RET

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

;

; WRITE IN FIRST LINE OF THE DISPLAY...

;

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

LINE1: CJNE R1,#00h,NEWADDR1

MOV R0,#80h ; CURSOR ADDRESS=

JMP SEND1 ; beginning of first line

NEWADDR1: MOV R0,#88h ; middle of line

SEND1: CALL send\_cmd

CALL WRITE ; Write the full line

RET

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

;

; WRITE IN SECOND LINE OF THE DISPLAY...

;

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

LINE2: CJNE R1,#00h,NEWADDR2

MOV R0,#0C0h ;set CURSOR ADDRESS

JMP SEND2

NEWADDR2: MOV R0,#0C8h

SEND2: CALL send\_cmd

CALL WRITE ;Write the full line

RET

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

;

; WRITE IN BOTH LINES OF THE DISPLAY...

;

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

BOTHLINES: PUSH PSW ;=>Save Carry

ACALL LINE1

INC DPTR

ACALL LINE2

FIN: POP PSW

RET

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

;

; LONG DELAY ROUTINE

;

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

LDELAY: ; delay 100ms

MOV R7,#10 ; 10 \* 500us = 5ms

DLY1: MOV R6,#229 ; 229 \* 2.17us = 500us

DJNZ R6,$ ; sit here for 500us

DJNZ R7,DLY1 ; repeat 200 times (100ms delay)

RET

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

;

; Subroutines for the temperature sensor

;

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

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

;

; TEMPERATURE

;

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

; This routine gets back the value of temperature from ADC1H.

; It compares ADC1H and 80H because 80H = 0 degC, +1LSB on ADC1H = +1 degC

; and -1LSB = -1 degC.

; If ADC1H > 80H -> SUP0 : positive temperature

; If ADC1h < 80H -> negative celsius temperature

TEMPER: MOV A,ADC1H

JB ACC.7,SUP0

CPL A

CLR ACC.7

INC A

MOV NEG,A

LCALL HEXTODEC

LCALL SAVEC

LCALL CELTOFAH

JMP RET1

SUP0: SUBB A,#80H

MOV NEG,#00H

LCALL HEXTODEC

LCALL SAVEC

MOV A,ADC1H

SUBB A,#80H

MOV TEMP,A

LCALL CELTOFAH

RET1: RET

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

;

; CELSIUS TO FAHRENHEIT

;

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

; this routine converts the celsius temperature in fahrenheit

; temperature.

; There are three cases: the temperature in celsius is

; - positive -> routine SUP00 : TEMP/5\*9+32

; - between -17 and 0 degC -> routine BETWEEN : 32-NEG/5\*9

; - lower -18 degC -> routine SUP17C : NEG/5\*9-32

CELTOFAH: MOV A,NEG

JZ SUP00C

CJNE A,#12H,SUITE

SUITE: JC BETWEEN

JNC SUP17C

SUP00C: MOV A,TEMP

MOV B,#05H

DIV AB

MOV R4,A

MOV A,#09H

MUL AB

MOV B,#05H

DIV AB

MOV R5,A

MOV A,R4

MOV B,#09H

MUL AB

ADD A,R5

ADD A,#20H

JMP FOLLOW

BETWEEN: MOV A,NEG

MOV B,#05H

DIV AB

MOV R4,A

MOV A,#09H

MUL AB

MOV B,#05H

DIV AB

MOV R5,A

MOV A,R4

MOV B,#09H

MUL AB

ADD A,R5

MOV R4,A

MOV A,#20H

SUBB A,R4

JMP FOLLOW

SUP17C: MOV A,NEG

MOV B,#05H

DIV AB

MOV R4,A

MOV A,#09H

MUL AB

MOV B,#05H

DIV AB

MOV R5,A

MOV A,R4

MOV B,#09H

MUL AB

ADD A,R5

SUBB A,#20H

FOLLOW: LCALL HEXTODEC

MOV A,DIG1

ADD A,#30H

MOV DIG1,A

MOV A,DIG2

ADD A,#30H

MOV DIG2,A

MOV A,DIG3

ADD A,#30H

MOV DIG3,A

RET

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

;

; HEXA TO DECIMAL

;

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

; This loop converts the number in A from HEX

; into the 3 decimal digits which represent the no.

HEXTODEC: CLR 00h

MOV R5, #0FFh

MOV R6, #0FFh

MOV R7, #0FFh

LOOP2: INC R5

CLR CY

SUBB A,#10

JNC LOOP2

CLR CY

ADD A,#10

MOV DIG1,A

MOV A,R5

LOOP3: INC R6

CLR CY

SUBB A,#10

JNC LOOP3

CLR CY

ADD A,#10

MOV DIG2,A

MOV A,R6

LOOP4: INC R7

CLR CY

SUBB A,#10

JNC LOOP4

CLR CY

ADD A,#10

MOV DIG3, A

RET

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

;

; SAVE CELSIUS DIGITS

;

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

; This routine saves the three digits of celsius temperature

; before a second use of HEXTODEC for fahrenheit temperature.

; Before to save, 30h is additions to have in DIGC1,2 and 3 the

; charater codes of the number to display.

SAVEC:

MOV A,DIG1

ADD A,#30H

MOV DIGC1,A

MOV A,DIG2

ADD A,#30H

MOV DIGC2,A

MOV A,DIG3

ADD A,#30H

MOV DIGC3,A

RET

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

;

; DISPLAY A MINUS (CELSIUS)

;

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

MINUSC: MOV R0,#87H

LCALL SEND\_CMD

MOV R0,#2DH

LCALL SEND\_CHAR

RET

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

;

; DISPLAY A MINUS (FAHRENHEIT)

;

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

MINUSF: MOV R0,#0C7H

LCALL SEND\_CMD

MOV R0,#2DH

LCALL SEND\_CHAR

RET

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

;

; CLEAR CHARRACTER

;

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

; Clear character on the diplay, the address of this

; character is in R0

CLRCHAR: LCALL SEND\_CMD

MOV R0,#20H

LCALL SEND\_CHAR

RET

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

;

; 2 SECONDES DELAY ROUTINE

;

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

WAIT: MOV SEC,#00H

MOV TIMECON,#01H ; Enable time clock

LOOP: MOV A,0A3H

CJNE A,#02H,LOOP ; Wait 2 secondes

RET

END