

MSP430 Based Digital Thermometer

***Using the Slope ADC of the Timer Port Module
to Measure Resistive Sensors***

*Application
Report*

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Contents

| | | |
|-------------------|---|------------|
| 1 | Introduction | 1 |
| 1.1 | Description of the MSP430 Ultra Low Power Microcontroller | 1 |
| 1.2 | Hardware Interfacing | 2 |
| 2 | Application Information | 4 |
| 3 | Timer Port Features | 5 |
| 4 | Summary | 5 |
| 5 | References | 5 |
| Appendix A | Software Listing | A-1 |

List of Figures

| | | |
|---|---|---|
| 1 | MSP430x32x Block Diagram | 2 |
| 2 | Digital Thermometer Circuit | 3 |
| 3 | Timer Port Module Application Example | 5 |

MSP430 Based Digital Thermometer

Brian Merritt

ABSTRACT

This application report describes a digital thermometer design that uses the slope ADC capabilities of the Timer Port module on the MSP430x3xx microcontrollers. This report can be used more generally as a reference on how to connect resistive sensors and reference resistors to the Timer Port module.

1 Introduction

This application report describes a digital thermometer design that uses the slope ADC capabilities of the Timer Port module on the MSP430x3xx microcontrollers. This report can be used more generally as a reference on how to connect resistive sensors and reference resistors to the Timer Port module.

All MSP430x3xx devices include the Timer Port module. The module allows several resistive sensors and reference resistors to be connected in an application. Unused module pins can be used as independent outputs.

1.1 Description of the MSP430 Ultra Low Power Microcontroller

The MSP430 is a 16-bit RISC-based microcontroller that uses advanced timing and design features, as well as a highly orthogonal structure, to deliver a processing core that is both powerful and very flexible. These features allow the MSP430 to consume only 400 μA in active mode in a typical 3-V system. The MSP430, typically using only 2 μA in standby mode, can wake up to fully synchronized active mode in a maximum of 6 μs . The MSP430 subfamilies incorporate various mixes of peripheral modules which result in highly integrated systems. Figure 1 shows a block diagram of the MSP430x32x.

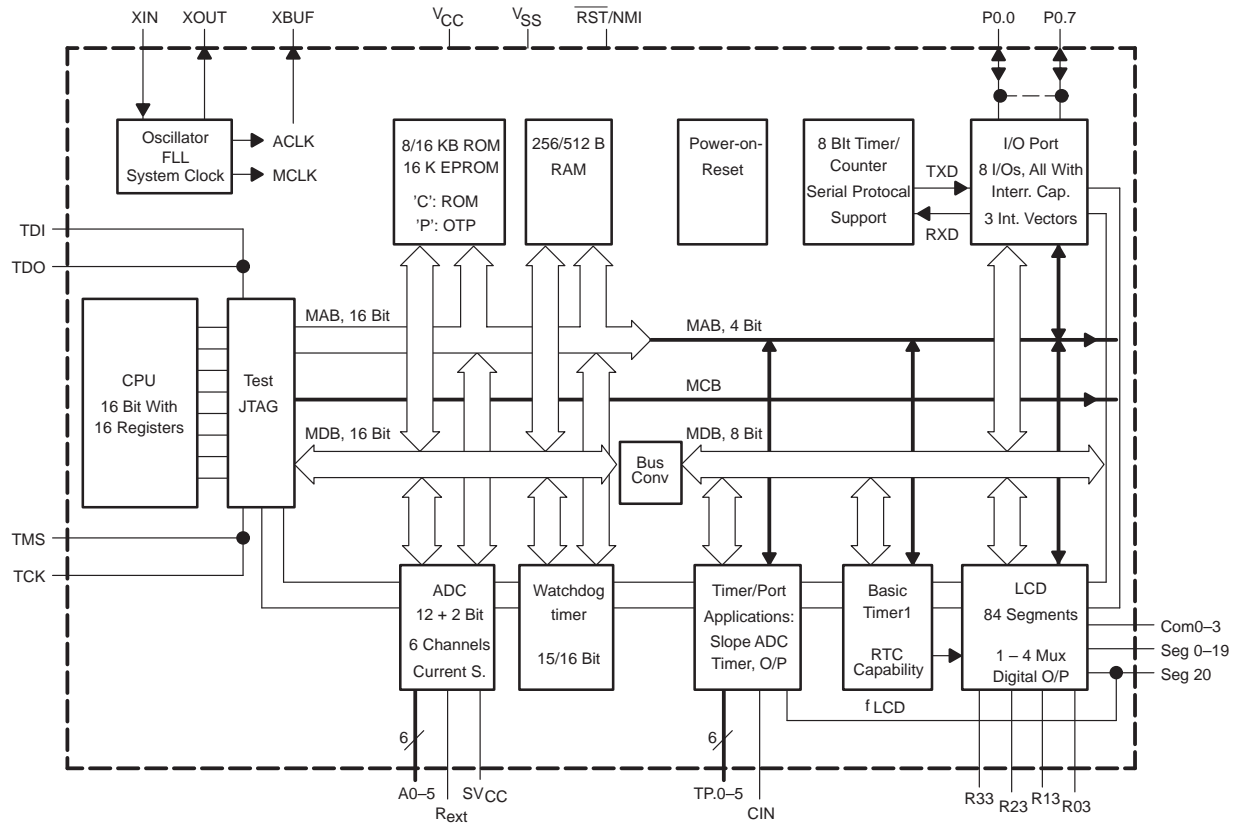


Figure 1. MSP430x32x Block Diagram

1.2 Hardware Interfacing

The hardware interface circuit is simply a thermistor (Radio Shack #271-110), a 10-k Ω reference resistor, and a 0.1- μ F capacitor. The components connect directly to the MSP430 as shown in Figure 2. An LCD display must also be connected if a visual readout of the measurements is desired.

The circuit performs a measurement by charging the capacitor to approximately V_{CC} , then discharging it through the reference resistor, while counting the number of internal clock cycles it takes until the CIN input goes low. The capacitor is charged to near V_{CC} again and then discharged through the thermistor, while counting the internal clock cycles required. The unknown resistance value of the thermistor can then be determined by taking a ratio of clock cycles required to discharge the capacitor via the thermistor, versus the number required to discharge via the known reference resistor value then multiplying the result by the value of the reference resistor. Software routines calculate the actual value of the thermistor, equate the value to a corresponding temperature, convert it to degrees Fahrenheit, and display the value on the LCD. Even though the last reading is constantly displayed, the MSP430 spends the majority of its time in low power mode 3 (LPM3). This time could be used to make additional measurements, to communicate with other components, or to perform calculations.

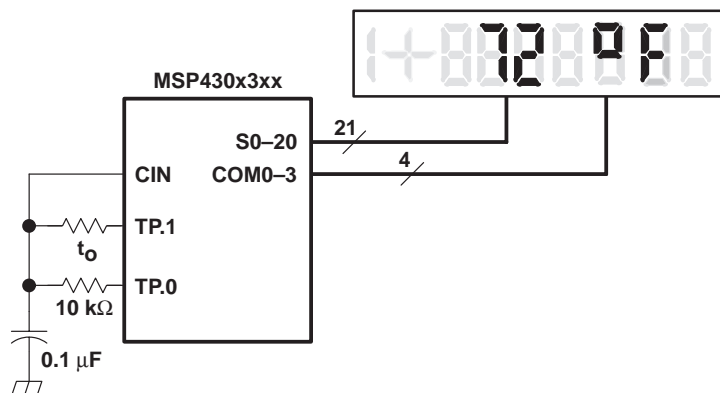


Figure 2. Digital Thermometer Circuit

The three components used to make the temperature measurement can be connected directly to a Texas Instruments MSP430 starter kit (STK) or evaluation kit (EVK). All of the other required connections, including those for the LCD, are already in place on the STK and EVK boards. The attached code is sized to fit completely into the 512 bytes of RAM memory that is available on the STK and EVK boards, which are based on the MSP430x325 devices. The code can be loaded into RAM through the serial port of a PC, using the interface included with the boards.

2 Application Information

The formula to measure the discharge time of the capacitor is:

$$t = -R \times C \times \ln \frac{V_{ref}}{V_{CC}}$$

$$t = N \times t_{clock} \quad (N \text{ is the number of clocks cycles})$$

$$N \times t_{clock} = -R \times C \times \ln \frac{V_{ref}}{V_{CC}}$$

$$N = -R \times C \times f_{clock} \times \ln \frac{V_{ref}}{V_{CC}}$$

The values of C , f_{clock} , and V_{ref}/V_{CC} are known. The value of the resistive sensor can be determined by the following formula, since the value of the reference resistor is a stable and known value.

$$\frac{N_{sensor}}{N_{ref}} = \frac{-R_{sensor} \times C \times f_{clock} \times \ln \frac{V_{ref}}{V_{CC}}}{-R_{ref} \times C \times f_{clock} \times \ln \frac{V_{ref}}{V_{CC}}}$$

$$\frac{N_{sensor}}{N_{ref}} = \frac{R_{sensor}}{R_{ref}}$$

$$R_{sensor} = R_{ref} \frac{N_{sensor}}{N_{ref}}$$

The formula resulting from the circuit in Figure 2 would be equal to:

$$R_{sensor} = 10 \text{ k}\Omega \frac{N_{sensor}}{N_{ref}}$$

3 Timer Port Features

The Timer Port module can support various configurations of resistive sensors and reference resistors. If several measurements are to be made in the same general range, then several sensors with only one reference resistor could be used. If measurements are made in ranges that are not relatively close, then sensors with individual reference resistors could be used (see Figure 3). Any unused pins can be used as digital outputs. The Timer Port module also has two 8-bit counters that can be cascaded to form one 16-bit counter. These counters may be used for other purposes when not being used by the Timer Port. See the *Metering Applications Report* (literature number SLAAE10C) and the *Architecture Guide and Module Library User's Guide* (literature number SLAUE10B) for additional information.

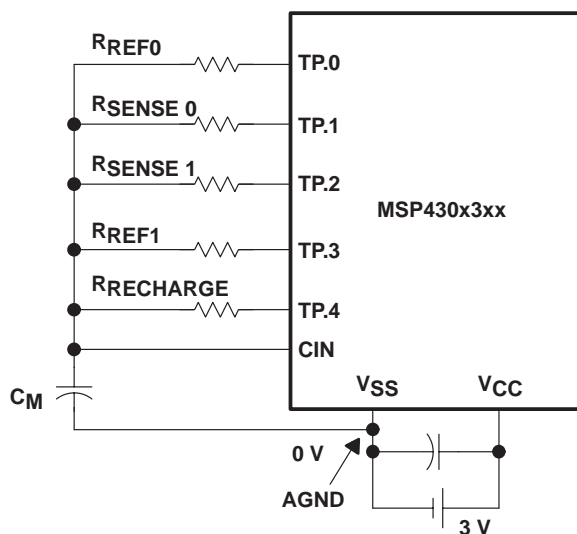


Figure 3. Timer Port Module Application Example

4 Summary

The Timer Port is a very versatile module that is available on MSP430x3xx microcontrollers. It is capable of supporting a wide variety of resistive sensor and reference resistor combinations. Components can be directly connected to the Timer Port to form complete sensor systems with a minimum of hardware interfacing. The combination of the Timer Port module, the 16-bit CPU, and the ultra low power design provide unmatched MIPS per watt performance.

5 References

1. *MSP430 Family Architecture Guide and Module Library User's Guide*, SLAUE10B, Texas Instruments Incorporated, 1996
2. *MSP430 Family Metering Applications Report*, SLAAE10C, 1998

Appendix A Software Listing

```

;*****
; DIGITAL THERMOMETER PROGRAM.
;*****
; THIS PROGRAM DEMONSTRATES THE USE OF THE TIMER PORT MODULE TO MAKE
; MEASUREMENTS OF RESISTIVE SENSOR VALUES. THE PROGRAM WILL RUN IN THE RAM
; MEMORY SPACE OF A MSP430 STK OR EVK DEVELOPMENT BOARD. THE PROGRAM CAN ALSO
; BE LOADED INTO ROM MEMORY ONCE THE "TOOL" BIT BELOW IS SET TO EQUAL 2. IF
; LOADED INTO ROM THE LOOKUP TABLE OF RESISTANCE VALUES CAN BE EXPANDED TO
; INCLUDE A WIDER TEMPERATURE RANGE.
;
; COMPONENTS OF THIS CODE WERE TAKEN FROM THE MSP430 METERING APPLICATIONS
; REPORT BY LUTZ BIERL, AND FROM EXAMPLE PROGRAMS WRITTEN BY MARK BUCCINI.
;
; TEMPDEMO VERSION 1.1, 4/1998
;
;*****
; SYSTEM DEFINITIONS FOR 320 STK/EVK
;*****
TOOL          .SET          0          ; 0 = STK/EVK RAM
                                           ; 1 = SIMULATOR
                                           ; 2 = ON-CHIP ROM

STACK         .EQU          003DEH    ; STACKPOINTER
RAM_ORIG      .EQU          00200H    ; FREE MEMORY START ADDRESS
ROM_ORIG      .EQU          0C100H    ; ROM START ON 320

               .IF          TOOL = 0
I_VECTORS     .EQU          003FFH    ; INTERRUPT VECTORS IN RAM
MAIN          .EQU          RAM_ORIG+20H ; PROGRAM RAM START ADDRESS
BTLOAD        .EQU          035H      ; LOAD ACTUAL 0.5 SECOND INTERRUPT

               .ELSEIF     TOOL = 1
I_VECTORS     .EQU          0FFFFH    ; INTERRUPT VECTORS IN ROM
MAIN          .EQU          ROM_ORIG   ; PROGRAM ROM START
BTLOAD        .EQU          011H      ; LOAD FAST INTERRUPT, NOT 1 SEC

               .ELSE
I_VECTORS     .EQU          0FFFFH    ; INTERRUPT VECTORS IN ROM
MAIN          .EQU          ROM_ORIG   ; PROGRAM ROM START
BTLOAD        .EQU          035H      ; LOAD ACTUAL 0.5 SECOND INTERRUPT
               .ENDIF
;*****
; DEFINITION SECTION FOR TIMER PORT ADC
;*****
TPCTL         .EQU          04BH      ; TIMER PORT CONTROL REGISTER (04BH)
TPSSEL0       .EQU          040H      ; CLK SOURCE 0=CMP, 1=ACLK (BIT 6 OF TPCTL)
ENB           .EQU          020H      ; CONTROLS EN1 OF TPCNT1
                                           ; 1(+ENA=1)=CMP (BIT 5 OF TPCTL)
ENA           .EQU          010H      ; CONTROLS EN1 OF TPCNT1
                                           ; 1(+ENB=1)=CMP (BIT 4 OF TPCTL)
EN1           .EQU          008H      ; ENABLE FOR TPCNT1 READ ONLY (BIT 3
                                           ; FO TPCTL)
RC2FG         .EQU          004H      ; RIPPLE CARRY TPCNT2 (BIT 2 OF TPCTL)
EN1FG         .EQU          001H      ; EN1 FLAG BIT (BIT 0 OF TPCTL)
TPIE         .EQU          004H      ; TIMER PORT INTERRUPT ENABLE (BIT 3 OF IE2)
TPCNT1        .EQU          04CH      ; COUNTER LOW BYTE
TPCNT2        .EQU          04DH      ; COUNTER HIGH BYTE
TPD           .EQU          04EH      ; TP DATA REGISTER (0-5=TP OUTPUT
                                           ; DATA, 6=CPON, 7=B16=2-8B OR 1-16B CNTR)

```

```

B16      .EQU      080H      ; SEPARATE TIMERS (0), OR 1-16 BIT
                                ; TIMER (1)
CPON     .EQU      040H      ; COMP OFF (0), COMP ON (1)
TPDMAX   .EQU      002H      ; BIT POSITION OUTPUT TPD.MAX
                                ; (2=BIT1=TPD.1)
TPE      .EQU      04FH      ; TP DATA ENABLE REGISTER (0-5=TPD
                                ; ENABLES, 6-7=TPCNT2 CLK)
MSTACK   .EQU      03D2H     ; RESULT STACK - 1ST WORD
PRESET   .EQU      0E8H      ; PRESET TPCNT2 FOR CHARGING OF C, COUNT
                                ; STOPS WHEN TPCNT2 OVERFLOWS, VALUE ALLOWS
                                ; CAP TO CHARGE FOR 6 RC TIME CONSTANTS
;*****
; CONTROL REGISTER DEFININTIONS
;*****
IE1      .EQU      0H        ; INTERRUPT ENABLE REGISTER 1
IE2      .EQU      01H       ; INTERRUPT ENABLE REGISTER 2
P01IE   .EQU      08H       ; P0.1 INTERRUPT ENABLE IN IE1
BTIE     .EQU      080H      ; BASIC TIMER INTERRUPT ENABLE IN IE2
IFG1     .EQU      02H       ; INTERRUPT FLAG REGISTER 1
IFG2     .EQU      03H       ; INTERRUPT FLAG REGISTER 2
LCDCTL   .EQU      030H      ; LCD CONTROL REGISTER
LCDM1    .EQU      031H      ; FIRST LCD DISPLAY MEM LOCATION
BTCTL    .EQU      040H      ; BASIC TIMER CONTROL REGISTER
BTCNT1   .EQU      0046H     ; BASIC TIMER COUNTER 1
BTCNT2   .EQU      0047H     ; BASIC TIMER COUNTER 2
WDTCTL   .EQU      0120H     ; WATCHDOG CONTROL REGISTER
WDTHOLD  .EQU      080H      ; PATTERN TO HOLD WATCHDOG
WDT_KEY  .EQU      05A00H    ; KEY TO ACCESS WATCHDOG
WDT_STOP .EQU      05A80H    ; WATCHDOG HOLD+KEY
;
GIE      .SET      8H        ; GENERAL INTERRUPT ENABLE
CPUOFF   .SET      10H       ; BIT TO TURN CPU OFF
OSCOFF   .SET      20H       ; BIT TO TURN OSCILLATOR OFF
SCG0     .SET      40H       ; SYS CLK GENERATOR CONTROL BIT 0
SCG1     .SET      80H       ; SYS CLK GENERATOR CONTROL BIT 1
LPM0     .SET      CPUOFF     ; BITS TO SET FOR LOW POWER MODE 0
LPM1     .SET      SCG0+CPUOFF ; " " " " " " " " 1
LPM2     .SET      SCG1+CPUOFF ; " " " " " " " " 2
LPM3     .SET      SCG1+SCG0+CPUOFF ; " " " " " " " " 3
LPM4     .SET      OSCOFF+CPUOFF ; " " " " " " " " 4
;*****
; REGISTERS USED TO SUPPORT CALCULATION OF SENSOR RESISTANCE
;*****
MLTPLR_HW .EQU      R5
TEN_K     .EQU      R6
BITTEST   .EQU      R7
MRESLT_HW .EQU      R8
MRESLT_LW .EQU      R9
LPCNTR    .EQU      R10
RESULT    .EQU      R11
;*****
;
; RESET PROGRAM
;*****
        .SECT "MAIN",MAIN
RESET    MOV        #STACK,SP    ; INITIALIZE STACKPOINTER
;*****
;
; SETUP UP PERIPHERALS
;*****
SETUP
SETUPINT MOV.B      #P01IE,&IE1 ; ENABLE P0.1/UART FOR RS232 MONITOR

```

```

MOV.B #BTIE+TPIE,&IE2 ; ENABLE B.TIMER, & TMR. PORT INTRPTS.
CLR.B &IFG1 ; CLEAR ANY INTERRUPT FLAGS
CLR.B &IFG2 ; CLEAR ANY INTERRUPT FLAGS
EINT ; ENABLE INTERRUPTS
SETUPWDT MOV #WDT_STOP,&WDTCTL ; STOP WATCHDOG TIMER
SETUPLCD MOV.B #0FFH,&LCDCTL ; STK LCD, ALL SEG, 4MUX
SETUPBT MOV.B #BTLOAD,&BTCTL ; LOAD BASIC TIMER WITH INTERRUPT FREQ
CLR.B &BTCNT1 ; CLEAR BT COUNTER 1
CLR.B &BTCNT2 ; CLEAR BT COUNTER 2
CLEARLCD MOV #15,R6 ; 15 LCD MEM LOCATIONS TO CLEAR
CLEAR1 MOV.B #0,LCDM1-1(R6) ; WRITE ZEROS IN LCD RAM LOCATIONS
DEC R6 ; ALL LCD MEM CLEAR?
JNZ CLEAR1 ; MORE LCD MEM TO CLEAR GO
;*****
; BEGIN MAIN PROGRAM
;*****

BEGIN BIS #LPM3,SR ; SET SR BITS FOR LPM3
;*****
; MEASUREMENT SUBROUTINE WITHOUT INTERRUPT. TP.2-.5 ARE NOT USED
; AND THEREFORE OVERRITTEN. ONLY TPD.0 & 1 USED.
; INITIALIZATION: STACK INDEX = 0, START WITH TPD.1
; 16-BIT TIMER, MCLK, CIN ENABLES COUNTING
;*****
MEASURE PUSH.B #TPDMAX ;PUSH TO STACK FOR LATER USE
CLR R8 ;INDEX FOR RESULT STACK
MEASLOP MOV.B #(TPSSEL0*3)+ENA,&TPCTL ;TPCNT1 CLK=MCLK, EN1=1
;*****
; CAPACITOR C IS CHARGED UP FOR >5 TAU. N-1 OUTPUTS ARE USED
;*****
MOV.B #B16+TPDMAX-1,&TPD ;1-16BIT COUNTER, SELECT CHARGE OUTPUTS
MOV.B #TPDMAX-1,&TPE ;ENABLE CHARGE OUTPUTS
MOV.B #PRESET,&TPCNT2 ;LOAD NEG. CHARGE TIME
BIS #CPUOFF,SR ;LOW POWER MODE TO SAVE POWER
MOV.B @SP,&TPE ;ENABLE ONLY ACTUAL SENSOR
CLR.B &TPCNT2
;*****
; SWITCH ALL INTERRUPTS OFF TO ALLOW NON-INTERRUPTED START OF
; TIMER AND CAPACITOR DISCHARGE
;*****
DINT ;DISABLE INTERRUPTS-ALLOW NEXT 2
CLR.B &TPCNT1 ;CLEAR LOW BYTE OF TIMER
BIC.B @SP,&TPD ;SWITCH ACTUAL SENSOR TO LOW
MOV.B #(TPSSEL0*3)+ENA+ENB,&TPCTL ;TPCNT1 CLK=MCLK, ENABLE CIN INPUT
EINT ;ENABLE INTERRUPTS-COMMON START
BIS #CPUOFF,SR ;CPU OFF TO SAVE POWER
;*****
; EN=0: END OF CONVERSION: STORE 2X8 BIT RESULT ON MSTACK
; ADDRESS NEXT SENSOR: IF NO OTHER SENSOR END REACHED
;*****
MOV.B &TPCNT1,MSTACK(R8) ;STORE RESULT ON STACK
MOV.B &TPCNT2,MSTACK+1(R8) ;STORE HIGH BYTE IN NEXT STACK BYTE
L$301 INCD R8 ;ADDRESS NEXT WORD
RRA.B @SP ;NEXT OUTPUT TPD.X
JNC MEASLOP ;IF C=1 - FINISHED
INCD SP ;HOUSEKEEPING-TPDMAX OFF STACK

```

```

;*****
; CALCULATE RESISTANCE OF SENSOR
;*****
; UNSIGNED MULTIPLY SUBROUTINE: MSTACK X TEN_K -> MRESLT_HW/MRESLT_LW
; USED REGISTERS MSTACK, TEN_K, MLTPLR_HW, MRESLT_LW, MRESLT_HW, BITTEST
; UNSIGNED MULTIPLY AND ACCUMULATE SUBROUTINE:
; (MSTACK X TEN_K) + MRESLT_HW|MRESLT_LW -> MRESLT_HW|MRESLT_LW
;*****
CALC_RES
      MOV     #10000,TEN_K           ; MOVE 10,000 DECIMAL INTO TEN_K
MPYU  CLR     MRESLT_LW             ; 0 -> LSBS RESULT
      CLR     MRESLT_HW             ; 0 -> MSBS RESULT
;
MACU  CLR     MLTPLR_HW             ; MSBS MULTIPLIER
      MOV     #1,BITTEST            ; BIT TEST REGISTER
L$002 BIT    BITTEST,MSTACK        ; TEST ACTUAL BIT
      JZ      L$01                  ; IF 0: DO NOTHING
      ADD     TEN_K,MRESLT_LW       ; IF 1: ADD MULTIPLIER TO RESULT
      ADDC   MLTPLR_HW,MRESLT_HW
L$01  RLA     TEN_K                 ; MULTIPLIER X 2
      RLC     MLTPLR_HW             ;
;
      RLA     BITTEST              ; NEXT BIT TO TEST
      JNC    L$002                 ; IF BIT IN CARRY: FINISHED
;*****
; UNSIGNED DIVISION SUBROUTINE 32-BIT BY 16-BIT
; REGISTERS USED (MSTACK+2), MRESLT_LW, RESULT, LPCNTR, MRESLT_HW
; MRESLT_HW MRESLT_LW / (MSTACK+2) -> RESULT REMAINDER IN MRESLT_HW
; RETURN: CARRY = 0: OK   CARRY = 1: QUOTIENT > 16 BITS
;*****
DIVIDE CLR    RESULT                ; CLEAR RESULT
      MOV     #17,LPCNTR            ; INITIALIZE LOOP COUNTER
DIV1  CMP     MSTACK+2,MRESLT_HW    ;
      JLO    DIV2                  ;
      SUB     MSTACK+2,MRESLT_HW
DIV2  RLC     RESULT                ;
      JC     RES_2_F               ; ERROR: RESULT > 16 BITS
      DEC     LPCNTR               ; DECREMENT LOOP COUNTER
      JZ      DIV3                 ; IS 0: TERMINATE W/O ERROR
      RLA     MRESLT_LW
      RLC     MRESLT_HW
      JNC    DIV1
      SUB     MSTACK+2,MRESLT_HW
      SETC
      JMP     DIV2
DIV3  CLRC                    ; NO ERROR, C = 0
;*****
; CONVERT RESISTANCE OF SENSOR TO DEGREES F FOR DISPLAY
;*****
RES_2_F
      CLR     R12                  ;POINTS TO VALUE IN RESISTANCE TABLE
      MOV     #064H,R13           ;MOVE MINIMUM TEMP-1 INTO TEMP INDICATOR
      JMP     FIRST_CMP           ;AVOID ADDING 1 ON FIRST COMPARE
CHECK_R  INCD    R12                ;INCREMENT RESISTANCE TABLE POINTER
      DADD   #1,R13                ;DECIMAL INCREMENT COUNTER
FIRST_CMP  CMP     RESIS_TAB(R12),RESULT ;COMPARE TABLE VALUE TO
      ; CALCULATED RESISTANCE
      JNC    CHECK_R              ;JUMP IF RSENSOR < TABLE VALUE @ POINTER

```

```

;*****
; DISPLAY "F" AND DEGREE SIGN ON LCD
;*****
DISPLAY      MOV.B  #A+E+F+G,LCDM1+1  ;"F" TO DISPLAY RAM
              MOV.B  #A+B+F+G,LCDM1+2  ;"o" (DEGREE SIGN) TO DISPLAY RAM

;*****
; DISPLAY BCD NUMBER IN R13 ON LCD
;*****
              MOV    R13,R12            ;COPY BCD NUMBER TO R12
              MOV    #LCDM1+4,R14       ;LOWER DIGIT LCD MEMORY LOCATION INTO R14
              BIC    #0FFF0H,R13        ;BLANK OFF ALL BUT LOWEST DIGIT
              MOV.B  LCD_TAB(R13),0(R14) ;LOWER DIGIT TO LCD
              MOV    R12,R13            ;REPLACE VALUE OF R13
              RRA    R13                 ;ROTATE VALUE RIGHT FOUR TIMES
              RRA    R13
              RRA    R13
              RRA    R13
              BIC    #0FFF0H,R13        ;BLANK OFF ALL BUT LOWEST DIGIT
              MOV.B  LCD_TAB(R13),1(R14) ;UPPER DIGIT TO LCD
              JMP    BEGIN               ;LOOP BACK TO BEGINNING OF PROGRAM
;*****
;          BASIC TIMER INTERRUPT SERVICE ROUTINE:
;          CPU IS RETURNED TO ACTIVE BY CLEARING LPM3 BITS ON SR ON THE STACK,
;*****
BTINT         BIC    #LPM3,0(SP)        ; CLEAR SR LPM3 BITS, ON TOP OF STACK
              RETI

;*****
;          TIMER PORT INTERRUPT SERVICE ROUTINE:
;          SYSTEM RETURNED TO ACTIVE ON RETI
;*****
TPINT         CLR.b  &TPCTL              ; CLEAR TP INTERRUPT FLAGS, TP OFF
              BIC    #LPM3,0(SP)        ; CLEAR SR LMP3 BITS, ON TOP OF STACK
              RETI

;*****
;          STK LCD TYPE
;*****
LCD_TYPE
A              .EQU  01H
B              .EQU  02H
C              .EQU  10H
D              .EQU  04H
E              .EQU  80H
F              .EQU  20H
G              .EQU  08H
H              .EQU  40H

LCD_TAB       .BYTE  A+B+C+D+E+F        ; DISPLAYS "0"
              .BYTE  B+C                ; DISPLAYS "1"
              .BYTE  A+B+D+E+G          ; DISPLAYS "2"
              .BYTE  A+B+C+D+G          ; DISPLAYS "3"
              .BYTE  B+C+F+G            ; DISPLAYS "4"
              .BYTE  A+C+D+F+G          ; DISPLAYS "5"
              .BYTE  A+C+D+E+F+G        ; DISPLAYS "6"
              .BYTE  A+B+C              ; DISPLAYS "7"
              .BYTE  A+B+C+D+E+F+G      ; DISPLAYS "8"
              .BYTE  A+B+C+D+F+G        ; DISPLAYS "9"

;*****

```

```

; RESISTANCE VALUES 65-99 DEGREES F.  VALUES = K OHMS X1000 - TO 3 DECIMAL PLACES
;*****
      .EVEN ; FOLLOWING SECTION MUST BE EVENLY ALIGNED
RESIS_TAB .WORD 12953 ;65 F
          .WORD 12666
          .WORD 12378
          .WORD 12090
          .WORD 11858
          .WORD 11626 ;70 F
          .WORD 11393
          .WORD 11161
          .WORD 10929
          .WORD 10697
          .WORD 10464 ;75 F
          .WORD 10232
          .WORD 10000
          .WORD 9813
          .WORD 9625
          .WORD 9438 ;80 F
          .WORD 9250
          .WORD 9063
          .WORD 8875
          .WORD 8688
          .WORD 8500 ;85 F
          .WORD 8313
          .WORD 8161
          .WORD 8008
          .WORD 7856
          .WORD 7703 ;90 F
          .WORD 7551
          .WORD 7398
          .WORD 7246
          .WORD 7093
          .WORD 6941 ;95 F
          .WORD 6817
          .WORD 6694
          .WORD 6570
          .WORD 6446 ;99 F
;*****
; INTERRUPT VECTORS
;*****
      .EVEN ; FOLLOWING SECTION MUST BE EVENLY ALIGNED
      .SECT "INT_VECT",I_VECTORS-31
      .WORD RESET ; PORT0, BIT 2 TO BIT 7
      .WORD BTINT ; BASIC TIMER
      .WORD RESET ; NO SOURCE
      .WORD RESET ; NO SOURCE
      .WORD RESET ; NO SOURCE
      .WORD TPINT ; TIMER PORT
      .WORD RESET ; NO SOURCE
      .WORD RESET ; NO SOURCE
      .WORD RESET ; NO SOURCE
      .WORD RESET ; NO SOURCE
      .WORD RESET ; WATCHDOG/TIMER, TIMER MODE
      .WORD RESET ; NO SOURCE
      .WORD RESET ; ADDRESS OF UART HANDLER
      .WORD RESET ; P0.0
      .WORD RESET ; NMI, OSC. FAULT
      .WORD RESET ; POR, EXT. RESET, WATCHDOG
      .END

```