Application Report
SLAA136A - October 2001

Ultrasonic Distance Measurement With the MSP430

Murugavel Raju *Mixed Signal Processors*

ABSTRACT

This application report describes a distance-measuring system based on ultrasonic sound utilizing the MSP430F413 ultralow-power microcontroller. The system transmits a burst of ultrasonic sound waves towards the subject and then receives the corresponding echo. The MSP430 integrated analog comparator Comparator_A is used to detect the arrival of the echo to the system. The time taken for the ultrasonic burst to travel the distance from the system to the subject and back to the system is accurately measured by the MSP430. Assuming the speed of sound in air at room temperature to be 1100 ft/s, the MSP430 computes the distance between the system and the subject and displays it using a two-digit static LCD driven by its integrated LCD driver. The distance is displayed in inches with an accuracy of ±1 inch. The minimum distance that this system can measure is eight inches and is limited by the transmitter's transducer settling-time. The maximum distance that can be measured is ninety-nine inches. The amplitude of the echo depends on the reflecting material, shape, and size. Sound-absorbing targets such as carpets and reflecting surfaces less than two square feet in area reflect poorly. The maximum measurable range is lower for such subjects. If the amplitude of the echo received by the system is so low that it is not detectable by the Comparator_A, the system goes out of range. This is indicated by displaying the error message *E*.

Contents

List of Figures

1 Theory of Operation

This application is based upon the reflection of sound waves. Sound waves are defined as longitudinal pressure waves in the medium in which they are travelling. Subjects whose dimensions are larger than the wavelength of the impinging sound waves reflect them; the reflected waves are called the echo. If the speed of sound in the medium is known and the time taken for the sound waves to travel the distance from the source to the subject and back to the source is measured, the distance from the source to the subject can be computed accurately. This is the measurement principle of this application. Here the medium for the sound waves is air, and the sound waves used are ultrasonic, since it is inaudible to humans.

Assuming that the speed of sound in air is 1100 feet/second at room temperature and that the measured time taken for the sound waves to travel the distance from the source to the subject and back to the source is *t* seconds, the distance *d* is computed by the formula *d=1100 X 12 X t inches*. Since the sound waves travel twice the distance between the source and the subject, the actual distance between the source and the subject will be *d/2*.

2 Circuit Description

The devices used to transmit and receive the ultrasonic sound waves in this application are 40-kHz ceramic ultrasonic transducers. The MSP430 drives the transmitter transducer with a 12-cycle burst of 40-kHz square-wave signal derived from the crystal oscillator, and the receiver transducer receives the echo. The Timer_A in the MSP430 is configured to count the 40-kHz crystal frequency such that the time measurement resolution is 25 µs, which is more than adequate for this application. The measurement time base is very stable as it is derived from a quartz-crystal oscillator. The echo received by the receiver transducer is amplified by an operational amplifier and the amplified output is fed to the Comparator_A input. The Comparator_A senses the presence of the echo signal at its input and triggers a capture of Timer_A count value to capture compare register CCR1. The capture is done exactly at the instant the echo arrives at the system. The captured count is the measure of the time taken for the ultrasonic burst to travel the distance from the system to the subject and back to the system. The distance in inches from the system to the subject is computed by the MSP430 using this measured time and displayed on a two-digit static LCD. Immediately after updating the display, the MSP430 goes to LPM3 sleep mode to save power. The Basic Timer1 is programmed to interrupt the MSP430 every 205 milliseconds. The interrupt signal from the Basic Timer1 wakes up the MSP430 to repeat the measurement cycle and update the display.

Figure 1 shows the circuit schematic diagram of this application. The MSP430F413 (U1) is the core of this system. Reference [1] is the data sheet for this device. LCD1 is a two-digit low-voltage static LCD driven by the integrated LCD driver. R03 is connected to V_{SS}, and R13 and R23 are left open for static-LCD-drive mode operation of the LCD peripheral. A 40-kHz crystal X1 is conveniently chosen for the low-frequency crystal oscillator to match the resonant frequency of the ultrasonic transducers used in this application. R12 serves as the pullup resistor for the reset line, and the integrated brownout-protection circuit takes care of brownout conditions. C9 provides power-supply decoupling to the MSP430 and is located close to the power supply lines of the device. A 14-pin box header (J1) allows JTAG interface to the MSP430 to provide in-circuit debugging and programming using the MSP430 flash emulation tool. LED1 is provided to indicate measurement cycles. Port pin P1.5 is configured to output the buffered 40-kHz square-wave ACLK required by the ultrasonic transmitter.

The output drive circuit for the transducer is powered directly from the 9-V battery and provides 18 V_{PP} drive to the ultrasonic transmitter. The 18 V_{PP} is achieved by a bridge configuration with hex inverter gates U4-CD4049. Reference [6] is the data sheet for this device. One inverter gate is used to provide a 180-degrees phase-shifted signal to one arm of the driver. The other arm is driven by the in-phase signal. This configuration doubles the voltage swing at the output and provides the required 18 V_{PP} to the transmitter transducer. Two gates are connected in parallel so that each arm can provide adequate current drive to the transducer. Capacitors C6 and C7 block the dc to the transducer. Since the CD4049 operates on 9-V and the MSP430 operates on a V_{CC} of 3.6 V, there is a logic level mismatch between the MSP430 and the output driver circuit. Bipolar transistor Q1 acts as a logic-level shifter between these two logic levels.

Operational amplifier U3 is the five-pin high-slew-rate TI operational amplifier TLV2771. Reference [5] is the data sheet for this device. This amplifier has a high-gain bandwidth and provides sufficiently high gain at 40 kHz. The operational amplifier is connected in an inverting amplifier configuration. R7 and R5 set the gain to 55 and C5 provides high-frequency rolloff. R3 and R4 bias the noninverting input to a virtual midrail for single-supply operation of the operational amplifier. The amplified ultrasonic signal swings above and below this virtual midrail. The high Q of transducer RX1 provides selectivity and rejection of unwanted frequencies other than 40 kHz. The output of the operational amplifier is connected to the Comparator_A CA0 input of the MSP430 via port pin P1.6. The Comparator_A reference is internally selected to be 0.5V_{CC}. When no ultrasonic echo is received, the voltage level at CA0 is slightly lower than the reference at CA1. When an echo is received, the voltage level increases above the reference and toggles the Comparator_A output CAOUT. R3 can be fine-tuned for the required sensitivity and the measurable range can be optimized.

The MSP430 and the ultrasonic signal amplifier circuit are powered by a regulated 3.6-V supply derived from the 9-V battery via TI LDO TPS77001. Reference [4] is the data sheet for this device. Resistors R1 and R2 program the regulator output voltage to 3.6 V. C1 and C2 are the recommended supply capacitors for correct functioning of the regulator. The transmitter driver is powered directly from the 9-V battery. Switch S1 functions as the power on switch for this application.

Figure 2 shows the oscilloscope trace of the 12-cycle, 40-kHz burst. Notice the 19.2-V peak to peak voltage swing. The ringing sine wave seen on top of the square waves is due to resonance in the transducer.

Figure 3 shows the oscilloscope traces for one complete measurement cycle. Trace 1 shows the 12-cycle, 40-kHz burst at the output of the transmitter transducer. Trace 2 shows the amplified receiver transducer output at pin 1 of the operational amplifier. The first burst-signal on the trace represents the signal directly received from the transmitter and is ignored by the MSP430. The next burst on the trace represents the echo reflected by the subject and is the signal used by the MSP430 for measurement. Trace 3 shows the width of the time interval measured by the MSP430. This width represents the time it takes for the burst to travel the distance from the measuring system to the subject and back, and it depends on the distance measured.

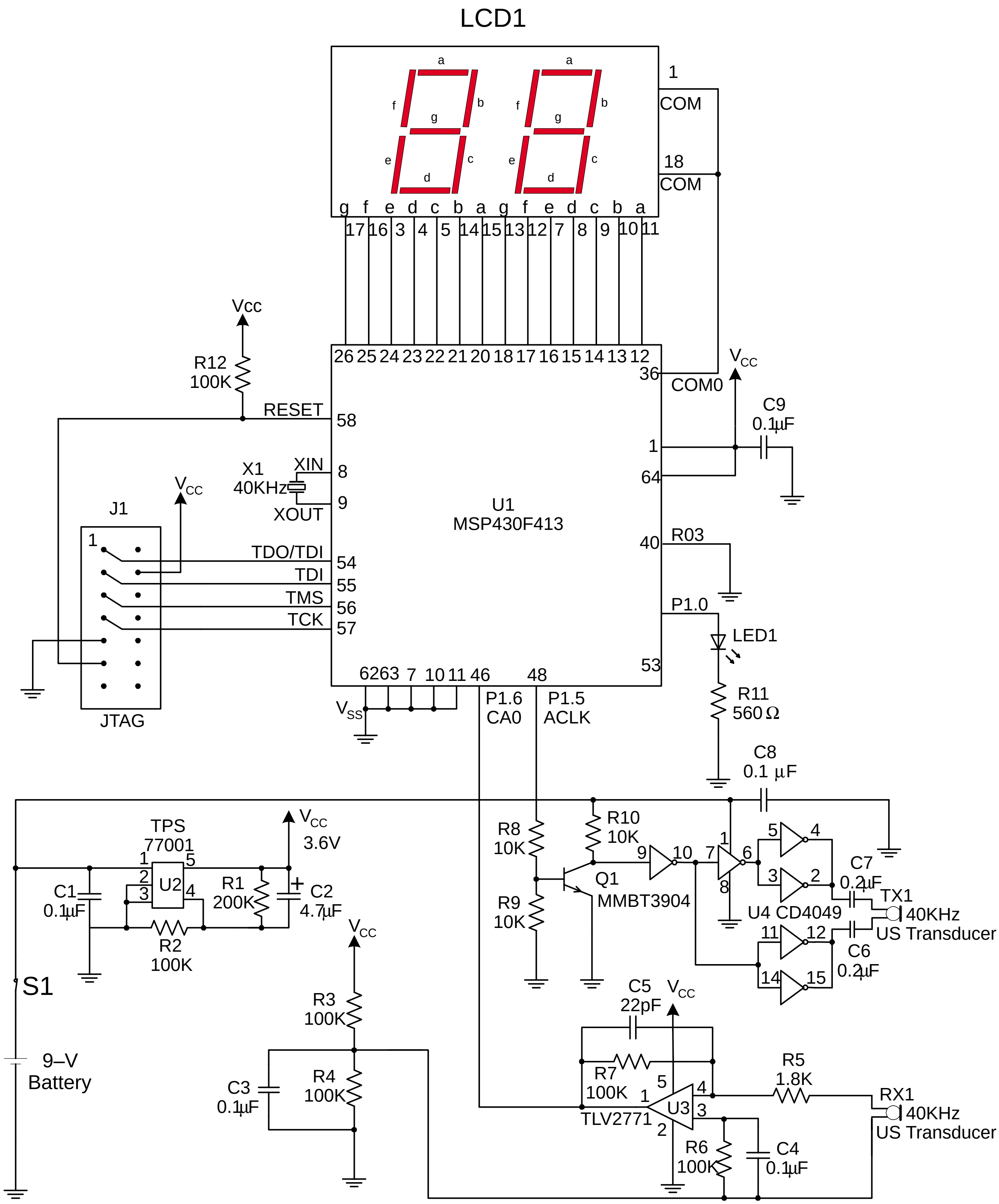

Figure 1. Circuit Schematic

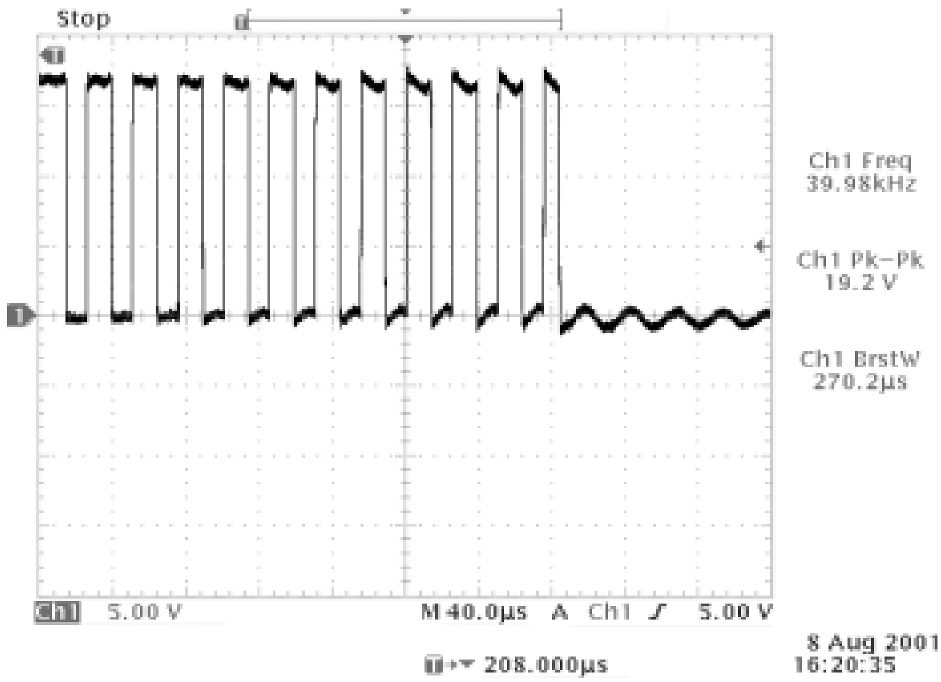

Figure 2. Oscilloscope Trace of Transmitter's 40-kHz Burst

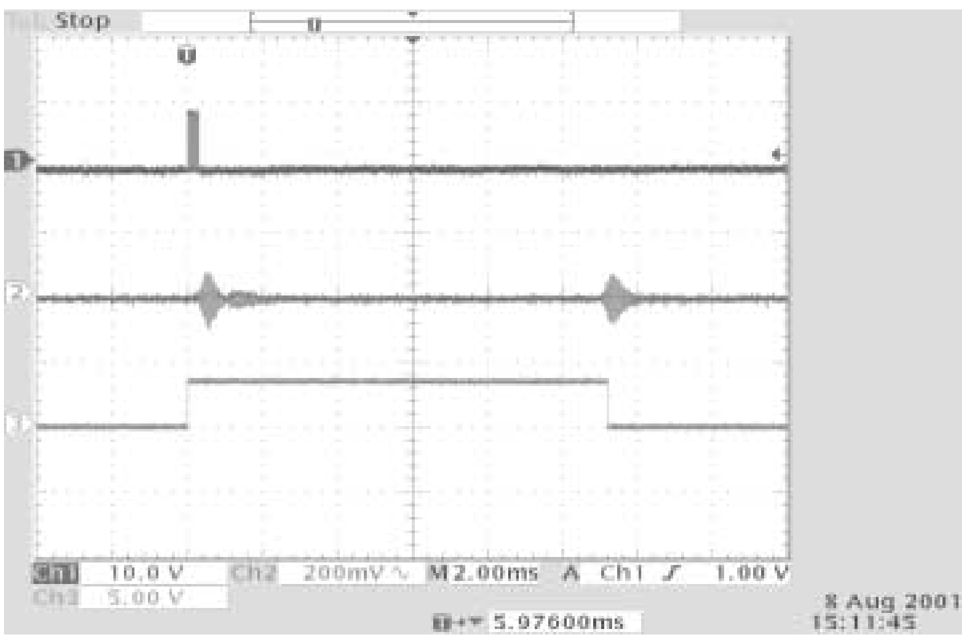

Figure 3. Oscilloscope Trace for One Measurement Cycle

3 Software Ultrasonic.s43

3.1 Init_Device

This subroutine initializes and configures the peripherals used. The Watchdog Timer is disabled first. A software delay is provided to allow the low-frequency oscillator to stabilize. The FLL+ multiplier is set to 64 to produce an MCLK frequency of 2.56 MHz. P1.0 is configured as an output for the LED. The unused port pins are configured as outputs and port pin P1.5 is configured to output the 40-kHz buffered ACLK frequency. The Basic Timer1 is enabled and configured to provide a 150-Hz LCD frequency and to interrupt the CPU every 205 milliseconds to initiate a measurement cycle. The Comparator_A is configured with $0.5V_{CC}$ internal reference and the CAPD bits are set to disable the input buffers for the comparator-input pins. The LCD module is turned on and configured for static-mode operation to drive the two-digit static LCD in the application. The LCD memory locations are cleared so that the initial LCD display is 00. The Basic Timer1 interrupt and the global interrupt enable are then enabled to allow the Basic Timer1 to periodically interrupt the CPU.

3.2 Mainloop

Mainloop updates the LCD with the value stored in the DIGITS buffer and then puts the MSP430 to LPM3 sleep mode. The MSP430 remains in sleep mode until a Basic Timer1 interrupt occurs and BT_ISR returns it to active mode. Now a measurement cycle is initiated. Timer_A is configured to 16-bit up mode and ACLK is selected as the clock source for Timer_A. CCR1 is set to the compare mode with a value of 12 so as to output a burst of 12 cycles of 40 kHz on P1.5. A 36-ACLK cycles delay follows to allow the output transducer to settle. This is realized by setting CCR1 to the compare mode with a value of 36. The MSP430 stays in LPM0 during these CCR1-compare wait states.

Now the system is set to receive the echo via the receiver transducer. The Comparator_A is configured to wait for the echo and it provides a capture interrupt at the instant the echo arrives. The Timer_A count is captured in capture-compare register CCR1. This value is the measure of the time it took the ultrasonic burst to travel the distance from the transmitter transducer to the subject and back to the receiver transducer. The count value is adjusted by adding 48 to compensate for the time lost in the 12-cycle burst and the 36-cycle transducer settling time delay. The adjusted value in CCR1 represents the exact time interval from the instant of the start of the burst to the instant of the start of the echo at the system. Next, the math subroutine is called to compute the actual distance in inches and return the result. If the system is out of range, the echo signal is not received and the Comparator_A does not provide a capture interrupt. The MSP430 stays in LPM0 until the next Basic Timer1 interrupt wakes it up. The CAIFG bit in the CCTL1 control register is then tested to make sure that the echo was never received. To indicate this condition, a value of *0xBE* is stored in DIGITS to display an *E* on the LCD. The program finally loops back to Mainloop to update the LCD and go back to LPM3 sleep mode. The next Basic Timer1 interrupt returns the MSP430 to active mode to repeat the program execution sequence.

3.3 Math_calc

The Math_calc subroutine takes care of the mathematical calculations required by this application. The adjusted 16-bit value from CCR1 is stored in the variable *Result*. This value is the representation of the time it takes the ultrasonic burst to travel the distance from the system to the subject and back to the system. Since Timer_A counts in 25-µs steps, the equivalent value in time will be *Result X 25 µS*. Assuming the speed of sound as 1100 ft/s at room temperature, the *Result* from the Timer_A count works out to be six counts per inch of distance. Therefore, dividing the *Result* by six produces the required value of the distance in inches.

To achieve the required precision with the available integer math of the MSP430, the 16-bit *Result* is first multiplied by 100 before dividing it by 6. This 16X16-bit multiplication is done by the subroutine Mul100. The 32-bit result is stored in the variables htX100_msw and htX100_lsw. This 32-bit result is then divided by 6 and the result is stored in the variable DIGITS. The value in DIGITS is in hexadecimal format. The hex2bcd subroutine converts this hexadecimal value to binary coded decimal (BCD) value, and the last two digits of the BCD number are discarded to compensate for the multiplication by 100 done earlier. The resulting two-digit value is returned to the variable DIGITS.

3.4 BT_ISR

The Basic Timer1 interrupt subroutine BT_ISR manipulates the bits in the status register SR residing in the stack such that the MSP430 returns to active mode on return from this ISR. This allows the MSP430 to continue to execute the code following the LPM3 instruction in Mainloop.

3.5 Display

This subroutine updates the two-digit static LCD with the value in the variable DIGITS. The segment data for the static display is stored in look-up table LCD_Tab. The LCD memory is loaded with the required segment data by correlating the numbers in DIGITS and indexing to the required location in the LCD_Tab look-up table.

3.6 Delay

This subroutine adds a 16-bit software delay. No registers are affected as the variable to be counted down by software is assigned to the top of stack (TOS). After the delay is timed out, the stack pointer (SP) is incremented back to the original value before returning from this subroutine.

4 Conclusion

The integrated analog Comparator_A, the 16-bit Timer_A with hardware capture/compare registers, the Basic Timer1, and the LCD driver peripherals simplify this ultrasonic distance measurement application design and provides a system-in-a-chip solution. The average current consumed by the application is 1.3 mA during a 15-inch distance measurement. This includes the quiescent current of LDO U2, operational amplifier U3, and CMOS hex inverter U4. The operational amplifier alone has a quiescent current of 1 mA and the remainder of the circuit current consumption is 300 µA. The LED draws 5 mA while it is on. The MSP430 draws an average current of 2.1 µA with the LCD continuously active. This is made possible by taking advantage of the ultralow-current features of the MSP430. The MSP430 sleeps in LPM3 most of the time and the CPU resources used by this application are only 5.6%.

Since the speed of sound is temperature dependent, the measured reading will be less accurate at temperatures other than room temperature. A simple thermistor-based temperature measurement and distance compensation could be employed in this application to allow the system to measure accurately over a wide range of temperatures. The measured distance and temperature data could also be stored in the flash memory if required. Adding additional receiver gain stages and using a multiplexed LCD to read out as many digits as required could increase the range.

5 References

1. *MSP430x41x Mixed Signal Microcontroller data sheet* SLAS340
2. *MSP430x4xx Family User's Guide*, SLAU056
3. *MSP430 Family Mixed-Signal Microcontrollers,* application report SLAA024
4. *TPS770xx Ultra Low-Power LDO Linear Regulators,* data sheet SLVS210
5. TLV277x Family of High-Slew-Rate *Operational Amplifiers*, data sheet SLOS209
6. *CD4049UB, CMOS Hex Inverting Buffer/Converter,* data sheet SCHS046A

Appendix A Programming Code

```
; THIS PROGRAM IS PROVIDED ”AS IS”. TI MAKES NO WARRANTIES OR
; REPRESENTATIONS, EITHER EXPRESS, IMPLIED OR STATUTORY,
; INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS
; FOR A PARTICULAR PURPOSE, LACK OF VIRUSES, ACCURACY OR
; COMPLETENESS OF RESPONSES, RESULTS AND LACK OF NEGLIGENCE.
; TI DISCLAIMS ANY WARRANTY OF TITLE, QUIET ENJOYMENT, QUIET
; POSSESSION, AND NON–INFRINGEMENT OF ANY THIRD PARTY
; INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO THE PROGRAM OR
; YOUR USE OF THE PROGRAM.
;
; IN NO EVENT SHALL TI BE LIABLE FOR ANY SPECIAL, INCIDENTAL,
; CONSEQUENTIAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ON ANY
; THEORY OF LIABILITY AND WHETHER OR NOT TI HAS BEEN ADVISED
; OF THE POSSIBILITY OF SUCH DAMAGES, ARISING IN ANY WAY OUT
; OF THIS AGREEMENT, THE PROGRAM, OR YOUR USE OF THE PROGRAM.
; EXCLUDED DAMAGES INCLUDE, BUT ARE NOT LIMITED TO, COST OF
; REMOVAL OR REINSTALLATION, COMPUTER TIME, LABOR COSTS, LOSS
; OF GOODWILL, LOSS OF PROFITS, LOSS OF SAVINGS, OR LOSS OF
; USE OR INTERRUPTION OF BUSINESS. IN NO EVENT WILL TI’S
; AGGREGATE LIABILITY UNDER THIS AGREEMENT OR ARISING OUT OF
; YOUR USE OF THE PROGRAM EXCEED FIVE HUNDRED DOLLARS
; (U.S.$500).
;
; Unless otherwise stated, the Program written and copyrighted
; by Texas Instruments is distributed as ”freeware”. You may,
; only under TI’s copyright in the Program, use and modify the
; Program without any charge or restriction. You may
; distribute to third parties, provided that you transfer a
; copy of this license to the third party and the third party
; agrees to these terms by its first use of the Program. You
; must reproduce the copyright notice and any other legend of
; ownership on each copy or partial copy, of the Program.
;
; You acknowledge and agree that the Program contains
; copyrighted material, trade secrets and other TI proprietary
; information and is protected by copyright laws,
; international copyright treaties, and trade secret laws, as
; well as other intellectual property laws. To protect TI’s
```

```
; rights in the Program, you agree not to decompile, reverse
; engineer, disassemble or otherwise translate any object code
; versions of the Program to a human-readable form. You agree
; that in no event will you alter, remove or destroy any
; copyright notice included in the Program. TI reserves all
; rights not specifically granted under this license. Except
; as specifically provided herein, nothing in this agreement
; shall be construed as conferring by implication, estoppel,
; or otherwise, upon you, any license or other right under any
; TI patents, copyrights or trade secrets.
;
; You may not use the Program in non-TI devices.
;
;*****************************************************************************
NAME         ULTRASONIC_DISTANCE_MEASUREMENT
;AUTHOR      Murugavel Raju
;            MSP430 Applications
;            Texas Instruments Inc.
;            Feb 2001
#include          "msp430x41x.h" ; Standard Equations
;
;*****************************************************************************
;    MSP430F413 Ultrasonic Distance Measurement Demonstration Program
;
;*****************************************************************************
;Register definitions
;*****************************************************************************
#define      DIGITS  R11
#define      Result  R10
#define      IRBT R9
#define      IROP1 R4
#define      IROP2L R5
#define      IROP2M R6
#define      IRACL R7
#define      IRACM R8
;*****************************************************************************
;Variables definition
;*****************************************************************************
             RSEG UDATA0
             htX100_msw: DS 2 ; word variable stored in RAM 200h & 201h
```

```
            htX100_lsw: DS 2 ;                              202h & 203h

;******************************************************************************
            RSEG    CSTACK                  ; Directive to begin stack segment
            DS      0
            RSEG    CODE                    ; Directive to begin code segment
RESET       mov.w   #SFE(CSTACK),SP         ; Define stack pointer
            call    #Init_Device            ; Initialize device
            mov.w   #0,DIGITS               ; Initialize DIGITS to '0'
Mainloop
            bic.b   #CAON,&CACTL1           ; Comparator_A OFF
            call    #Display                ; Display Data on LCD
            bis.w   #LPM3,SR                ; Wait in LPM3

;***************Start Ultrasonic Bursts and take measurements *****************


            clr.w   &CCTL1                  ; Disable CCTL1
            clr.w   &TACTL                  ; Disable timer_A
            bis.b   #BIT0,&P1OUT            ; LED ON
SetupTimerA mov.w   #TASSEL0+TACLR+MC1,&TACTL
                                            ; TACLK = ACLK, 16 bit up mode
            bis.b   #BIT5,&P1SEL            ; ACLK o/p on P1.5
            mov.w   #12,&CCR1               ; 12 cycle 40KHz burst
            mov.w   #CCIE,&CCTL1            ; Compare mode,interrupt
            bis.w   #LPM0,SR                ; Wait for CCR1 interrupt
            bic.b   #BIT5,&P1SEL            ; ACLK o/p on P1.5 OFF
TimerCLR    bis.w   #TACLR,&TACTL
            mov.w   #36,&CCR1               ; Delay for transducer to settle

            mov.w   #CCIE,&CCTL1            ; Compare mode,interrupt
            bis.w   #LPM0,SR                ; Wait for CCR1 interrupt

            bis.b   #CAON,&CACTL1           ; Comparator_A ON
            bic.b   #CAIFG,&CACTL1          ; Enable Comparator_A interrupt flag
            mov.w   #CM0+CCIS0+SCS+CAP+CCIE,&CCTL1
                                            ; Pos edge, CCIB,Cap,interrupt
            push    &TAR                    ; TOS = TAR at Start of measurement
            bis.w   #LPM0,SR                ; Wait for CCR1 interrupt (Echo)
            clr.w   &CCTL1                  ; Disable CCTL1
            bic.b   #BIT0,&P1OUT            ; LED OFF
            bit.b   #CAIFG,&CACTL1          ; Check for Echo not received
```

```
            jz      Next                    ; 'out of range' condition
            mov.w   &CCR1,Result            ; Result = TAR (CCR1) at EOC
            sub.w   @SP+,Result             ; Result = time taken
            add.w   #48,Result              ; compensate 12Clks for the burst
                                            ; transmission time + 36Clks delay

;***************** Measurement Done *****************************************

            call    #Math_calc              ; Call Math subroutine
            swpb    DIGITS                  ; Shift left by two digits for /100

            jmp     Mainloop                ; next measurement cycle
Next        mov.w   #0beh,DIGITS            ; No echo received display 'E' error
            jmp     Mainloop

;****************************************************************************
Init_Device                                 ; Initialize MSP430x41x
;****************************************************************************
            mov.w   #WDTPW+WDTHOLD,&WDTCTL  ; Stop WDT
            bis.b   #030h,&FLL_CTL0         ; Turn on internal load capacitors
                                            ; for the XTAL to start oscillation

            call    #Delay                  ; Delay for oscillator to stabilize
            mov.b   #03fh,&SCFQCTL          ; MCLK = 40KhzX64 = 2.56Mhz

            call    #Delay                  ; Delay for FLL to stabilize
SetupP1     mov.b   #000h,&P1OUT            ; Clear P1 output register
            bis.b   #0bfh,&P1DIR            ; Unused pins as o/p's
            bis.b   #040h,&P1SEL            ; Comp_A + i/p function

SetupP2     mov.b   #000h,&P2OUT            ; Clear P2 output register
            bis.b   #0ffh,&P2DIR            ; Unused pins as o/p's

SetupP6     mov.b   #000h,&P6OUT            ; Clear P6 output register
            bis.b   #0ffh,&P6DIR            ; Unused pins as o/p's
SetupBT     mov.b   #BTFRFQ0+BTFRFQ1+BTIP2+BTDIV,&BTCTL
                                            ; Enable BT with 150Hz LCD freq.
                                            ; and 205 millisecond interrupt
SetupCA     mov.b   #CAPD6,&CAPD            ; o/p buffer disable for comp i/p

            mov.b   #P2CA0,&CACTL2          ; P1.6 to Comp + input
            mov.b   #CARSEL+CAREF1+CAON,&CACTL1
```

```
                                        ; Comp_A ON, 0.5Vcc int. reference
SetupLCD     bis.b   #LCDON+LCDSON+LCDSG0_7,LCDCTL
                                        ; LCD module ON and in static mode
ClearLCD     mov     #15,R15            ; 15 LCD mem locations to clear
             mov.b   #LCDMEM,R14
Clear1       mov.b   #0,0(R14)          ; Write zeros in LCD RAM locations
             inc.b   R14
             dec     R15                ; All LCD mem clear?
             jnz     Clear1             ; More LCD mem to clear go

             bis.b   #BTIE,&IE2         ; Enable Basic Timer interrupt

             eint                       ; Enable interrupts

             ret
;******************************************************************************
BT_ISR                                  ; Basic Timer ISR, CPU returns
                                        ; to active mode on RETI
;******************************************************************************

             bic     #LPM3,0(SP)        ; Clear LPM3 bits on TOS
             reti                       ; on return from interrupt

;******************************************************************************
TAX_ISR;     Common ISR for CCR1–4 and overflow
;******************************************************************************
             add.w   &TAIV,PC           ; Add TA interrupt offset to PC
             reti                       ; CCR0 – no source
             jmp     CCR1_ISR           ; CCR1
             reti                       ; CCR2
             reti                       ; CCR3
             reti                       ; CCR4
TA_over      reti                       ; Timer_A overflow
CCR1_ISR     bic.w   #CCIFG,&CCTL1
             bic.w   #LPM0,0(SP)        ; Exit LPM0 on reti
             reti                       ;

;******************************************************************************
Display      ;Subroutine to Display values DIGIT1 & DIGIT2
             ;CPU Registers used R15, R14, R13 and R12,  not saved
;******************************************************************************
             mov.w   #LCDM1,R15         ; R15 points to first LCD location
```

```
            mov.b   DIGITS,R14              ; LSD value moved to R14
OutLCD      mov.b   R14,R13                 ; Copy value in R14 to R13
            rra.b   R13                     ; Right Shift
            rra.b   R13                     ; four times to
            rra.b   R13                     ; swap
            rra.b   R13                     ; nibbles
            and.b   #0Fh,R14                ; low nibble now in R14
            and.b   #0Fh,R13                ; high nibble now in R13
            mov.b   LCD_Tab(R14),R12        ; Low nibble to LCD digit 1
            mov.b   R12,0(R15)              ; Low nibble segments a & b to LCD
            rra.w   R12
            inc.b   R15
            mov.b   R12,0(R15)              ; Low nibble segments c & d to LCD
            rra.w   R12
            inc.b   R15
            mov.b   R12,0(R15)              ; Low nibble segments e & f to LCD
            rra.w   R12
            inc.b   R15
            mov.b   R12,0(R15)              ; Low nibble segments g & h to LCD
            rra.w   R12
            inc.b   R15
            mov.b   LCD_Tab(R13),R12        ; High nibble to LCD digit 2
            mov.b   R12,0(R15)              ; High nibble segments a & b to LCD
            rra.w   R12
            inc.b   R15
            mov.b   R12,0(R15)              ; High nibble segments c & d to LCD
            rra.w   R12
            inc.b   R15
            mov.b   R12,0(R15)              ; High nibble segments e & f to LCD
            rra.w   R12
            inc.b   R15
            mov.b   R12,0(R15)              ; High nibble segments g & h to LCD
            rra.w   R12
            ret
;*******************************************************************************
;           LCD Type Definition
;*******************************************************************************
;Segments definition
a           equ     001h
b           equ     010h
```

```
c             equ     002h
d             equ     020h
e             equ     004h
f             equ     040h
g             equ     008h
h             equ     080h
Blank         equ     000h

LCD_Tab       db      a+b+c+d+e+f             ; Displays ”0”
              db      b+c                     ; Displays ”1”
              db      a+b+d+e+g               ; Displays ”2”
              db      a+b+c+d+g               ; Displays ”3”
              db      b+c+f+g                 ; Displays ”4”
              db      a+c+d+f+g               ; Displays ”5”
              db      a+c+d+e+f+g             ; Displays ”6”
              db      a+b+c                   ; Displays ”7”
              db      a+b+c+d+e+f+g           ; Displays ”8”
              db      a+b+c+d+f+g             ; Displays ”9”
              db      a+b+c+e+f+g             ; Displays ”A”
              db      Blank                   ; Displays  Blank
              db      a+d+e+f                 ; Displays ”C”
              db      b+c+d+e+g               ; Displays ”D” d
              db      a+d+e+f+g               ; Displays ”E”
              db      a+e+f+g                 ; Displays ”F”
;*******************************************************************************
Delay;        Software delay
;*******************************************************************************
              push    #0FFFFh                 ; Delay to TOS
DL1           dec.w   0(SP)                   ; Decrement TOS
              jnz     DL1                     ; Delay over?
              incd    SP                      ; Clean TOS
              ret                             ; Return from subroutine

;*******************************************************************************
Math_calc;    calculation subroutine
;*******************************************************************************
              mov.w   #0h, DIGITS             ; Initialize DIGIT to 0
              cmp.w   #0h, Result             ; Check if Result count=0
              jeq     calc_over               ; Exit if 0
              call    #Mul100                 ; Multiply Result count by 100
```

```
            call    #Divide                   ; Divide the result with #06d
            call    #Hex2bcd                  ; Convert 16bit binary to BCD number
                                              ; Result xx.xx

calc_over   ret                               ; Return from subroutine

;*****************************************************************************
Mul100      ;subroutine for multiplying Result with 100d
            ;inputs Result 16bit and constant 64h (100d) 16bit
            ;output 32bit htX100_msw & htX100_lsw
;*****************************************************************************

            mov.w   #100,IROP1                ; Load IROP1 with 100 (multiplier)
mpyu        clr.w   htX100_lsw                ; Clear buffer for least
                                              ; Significant word
            clr.w   htX100_msw                ; Clear buffer for most
                                              ; Significant word
macu        clr.w   IROP2M                    ; Clear multiplier high word
L$002       bit.w   #1,IROP1                  ; Test actual bit
            jz      L$01                      ; If 0: do nothing
            add.w   Result,htX100_lsw         ; If 1: Add multiplier to Result
            addc.w  IROP2M,htX100_msw         ;
L$01        rla.w   Result                    ; Multiplier X 2
            rlc.w   IROP2M                    ;
            rrc.w   IROP1                     ; Next bit to test
            jnz     L$002                     ; If bit in carry : finished

            ret
;*****************************************************************************
Divide      ;Subroutine for 32/16 bits division
            ;inputs 32bit htX100_msw & htX100_lsw and #06 16bit, output DIGIT
16bit
;*****************************************************************************
            clr.w   DIGITS                    ; Clear buffer to hold new Result
            mov.w   #17,IRBT                  ; Initialize loop counter
div1        cmp.w   #06,htX100_msw            ; Compare divisor with dividend high
word
            jlo     div2                      ; If less : jump to div2
            sub.w   #06,htX100_msw            ; Subtract 6 from high word
div2        rlc.w   DIGITS                    ; Rotate result left through carry 1
bit
            jc      div4                      ; If carry set: finished
            dec.w   IRBT                      ; Decrement bit counter
```

```
            jz      div3                ; If counter = 0 : finished
            rla.w   htX100_lsw          ; Dividend X 2
            rlc.w   htX100_msw          ;
            jnc     div1                ; If carry not set jump to step div1
            sub.w   #06,htX100_msw      ; Subtract 6 from high word
            setc                        ; Set carry
            jmp div2                    ; Jump to repeat
div3        clrc                        ; Clear carry
div4        ret                         ; Return from subroutine
;*******************************************************************************
Hex2bcd     ;Subroutine for converting 16bit hexadecimal value to BCD value
            ;input in DIGITS 16bit hexadecimal, output in DIGITS 16bit BCD
;*******************************************************************************
            mov #16,r9                 ; R9 no of bits
            clr r8                     ; Clear R8
            clr r7                     ; Clear R7
L$1         rla DIGITS                 ; Rotate left arithmetic DIGITS
            dadd r7,r7                 ; Add source and carry decimally
            dadd r8,r8                 ; to destination
            dec r9                     ; Decrement bit counter
            jnz L$1                    ; Is 16 bits over ?
            mov r7,DIGITS              ; Result in DIGITS
            ret                        ; Return from subroutine

;*******************************************************************************
            COMMON  INTVEC              ; MSP430x41x Interrupt vectors

;*******************************************************************************

            ORG     BASICTIMER_VECTOR
BT_VEC      DW      BT_ISR              ; Basic Timer Vector
            ORG     TIMERA1_VECTOR      ; Timer_AX Vector
TIMA_VEC    DW      TAX_ISR             ;
            ORG     RESET_VECTOR
RESET_VEC   DW      RESET               ; POR, ext. Reset, Watchdog
;*******************************************************************************
            END
Title
```

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

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

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265