



MSP430 超声波测距

摘要

本应用报告描述了一种采用 MSP430F413 超低功耗微控制器的基于超声波的距离测量系统。系统向测量目标发射超声波脉冲然后接收相应的反射波。MSP430 集成的模拟比较器 A 用于检测到达系统的回声。超声波脉冲从系统到目标然后反射回系统所需的时间可以由 MSP430 精确的测量。假设声波室温下在空气中的速度为 1100 英尺/秒，MSP430 计算系统与目标间的距离并采用内部集成的 LCD 驱动器将其显示在一两位的静态液晶显示器上。距离以英寸为单位显示，精度为 ± 1 英寸。本系统能够测量的最小距离是 8 英寸，因为受到发射器传感器稳定时间；能够测量的最大距离为 99 英寸。回声的强度取决于反射物的材质、形状和尺寸。地毯之类的吸音材料或者反射面积小于两平方英尺的物体反射能力很弱，这类目标最大测量范围相对较小。如果系统接收到的回声的强度很小以至于比较器 A 无法检测到，就超出了系统的测量范围。这时系统将显示错误信息“E”。



内 容

1	工作原理.....	2
2	电路描述.....	2
3	软件 Ultrasonic.s43.....	6
3.1	设备初始化子程序.....	6
3.2	主循环程序.....	6
3.3	计算子程序.....	7
3.4	BT_ISR 子程序.....	7
3.5	显示子程序.....	7
3.6	延时子程序.....	7
4	结论.....	7
5	参考文献.....	8
A	程序代码.....	9

图表清单

1	电路原理图.....	4
2	发射器 40KHz 脉冲波形轨迹图.....	5
3	一个测量周期的波形轨迹图.....	5



1 工作原理

本应用基于声波的反射。声波在其传播的介质中被定义为纵波。当声波受到尺寸大于其波长的目标物阻挡时就会发生反射；反射波称为回声。如果声波在介质中传播的速度是已知的，而且测量到声波从声源到达目标然后返回声源的时间，从声源到目标的距离就可以精确地计算出来。这就是本应用的测量原理。这里声波传播的介质就是空气，采用不可见的超声波。

假设室温下声波在空气中的传播速度是 1100 英尺/秒，测量到的声波从声源到达目标然后返回声源的时间是 t 秒，距离 d 可以由下列公式计算：

$$d=1100 \times 12 \times t \text{ (英寸)}$$

因为声波经过的距离是声源与目标之间距离的两倍，声源与目标之间的实际距离应该为 $d/2$ 。

2 电路描述

本应用中用来发射和接收超声波的装置是 40KHz 的陶瓷超声波传感器。MSP430 采用晶体振荡器产生的一个 40KHz 方波信号的 12 周期脉冲序列驱动发射传感器，接收传感器则接收回声。MSP430 的定时器 A 被定义为 40KHz 晶振频率计数器，因此时间测量精度为 25 微秒，这对本应用来说足够用了。测量的时间基准是非常稳定的因为它由石英晶体振荡器产生。接收传感器收到的回声由运算放大器放大后送到比较器 A 的输入端。比较器 A 检测到其输入端的回声信号后触发定时器 A 计数值的捕捉来捕获比较寄存器 CCR1 的值。捕捉在回声到达系统的瞬间进行。捕捉的计数值就是超声波脉冲序列从系统出发到达目标然后返回系统的时间。从系统到目标的距离（用英寸表示）就可由 MSP430 用测得的时间算出并显示在两位静

态液晶显示器上。显示更新后，MSP430 马上转入 LPM3 睡眠模式来降低功耗。基本定时器 1 被编程用来每 205 毫秒中断 MSP430 一次。基本定时器 1 的中断信号唤醒 MSP430 并重复测量周期，更新显示。

图 1 显示的是本应用的电路原理图。其中 MSP430F413 (U1) 是系统的核心部分。参考文献[1]是这个芯片的技术资料。LCD1 是一由 MSP430F413 内部集成的 LCD 驱动器驱动的两位低压静态液晶显示器。R03 连接到 Vss, R13 和 R23 悬空，将 LCD 外围电路设置为静态 LCD 驱动模式。这里方便的选用了—个 40KHz 的低频晶振，与本应用中采用的超声波传感器的谐振频率相匹配。R12 作为复位线的上拉电阻，内部集成的掉电保护电路可以预防掉电情况。C9 位于靠近芯片电源线的位置，提供与 MSP430 的电源耦合。14 脚的接插件 (J1) 提供 JTAG 接口与 MSP430 相连，可使用 MSP430 FLASH 仿真工具进行在线调试和编程。LED1 用来指示测量周期。端口引脚 P1.5 被定义为输出超声波发射器所需要的 40KHz 方波 ACLK 信号。

传感器的输出驱动电路直接由 9V 电池供电并提供 18Vpp 驱动超声波发射器。18Vpp 是通过—个二进制非门 CD4049(U4)桥电路实现的。参考文献 [6]是它的技术资料。其中—个非门用来为驱动器的一侧提供 180 度的相移信号。另一侧由相内信号驱动。这种结构使输出端的电压提高了一倍，为发射传感器提供了 18Vpp 电压。两个门并联连接以便每一侧能够为传感器提供足够的驱动电流。电容 C6、C7 阻断了到传感器的直流通路。因为 CD4049 工作于 9V 而 MSP430 工作于 Vcc 3.6V，MSP430 和输出驱动器之间的逻辑电平是不匹配的，双极性晶体管 Q1 就作为这两种逻辑电平之间的转换器。

运算放大器 U3 是 TI 公司的高速运算放大器 TLV2771。参考文献[5]是它的技术资料。这个放大器具有高增益带宽并在 40KHz 时提供充分的高增益。运算放大器连接成反相放大器构造。R7、R5 设置增益为 55，C5 提供高频滚降。R3、R4 偏置非反相输入端，为运算放大器的单输入工作提供一个虚拟中间值。放大后的超声波信号在这个虚拟中间值上下波动。传感器 RX1 的高 Q 值提供选择性并丢弃除了 40KHz 之外的频率。运算放大器的输出端连接到比较器 A 的输入端 CA0（即端口引脚 P1.6）。比较器 A 的参考电平内部选择为 0.5V_{cc}。当接收到回声时，电压高于参考电平从而触发比较器 A 的输出 CAOUT。调整 R3 可以得到需要的灵敏度，并优化测量范围。

MSP430 和超声波信号放大器电路由 9V 电池经 TI 公司的 LDO 芯片 TPS77001 调制后的 3.6V 电压供电。参考文献[4]是它的技术资料。电阻 R1 和 R2 调整稳压器输出电压为 3.6V。C1 和 C2 是推荐使用的稳压器的正常工作的供电电容。发射器的驱动电路是由 9V 电池直接供电的。开关 S1 是本应用的电源开关。

图 2 显示的是 12 周期的 40KHz 脉冲序列的波形轨迹图。我们注意到 19.2V 的峰—峰电压波动。方波顶部的正弦响铃波是由于传感器内部的谐振引起的。

图 3 显示的是一个完整测量周期的波形轨迹。轨迹 1 显示的是发射传感器输出端的 12 周期的 40KHz 脉冲序列。轨迹 2 显示的是接收传感器输出经运算放大器放大后在引脚 1 上的输出。轨迹上的第一个脉冲序列信号代表直接从发射器上收到的信号，被 MSP430 忽略。接下去的脉冲序列代表目标反射的回声，被 MSP430 用于测量。轨迹 3 显示的是 MSP430 测得的时

间间隔的宽度。这个宽度代表该脉冲序列从系统到达目标再返回所花的时间，显然它取决于所测量的距离。

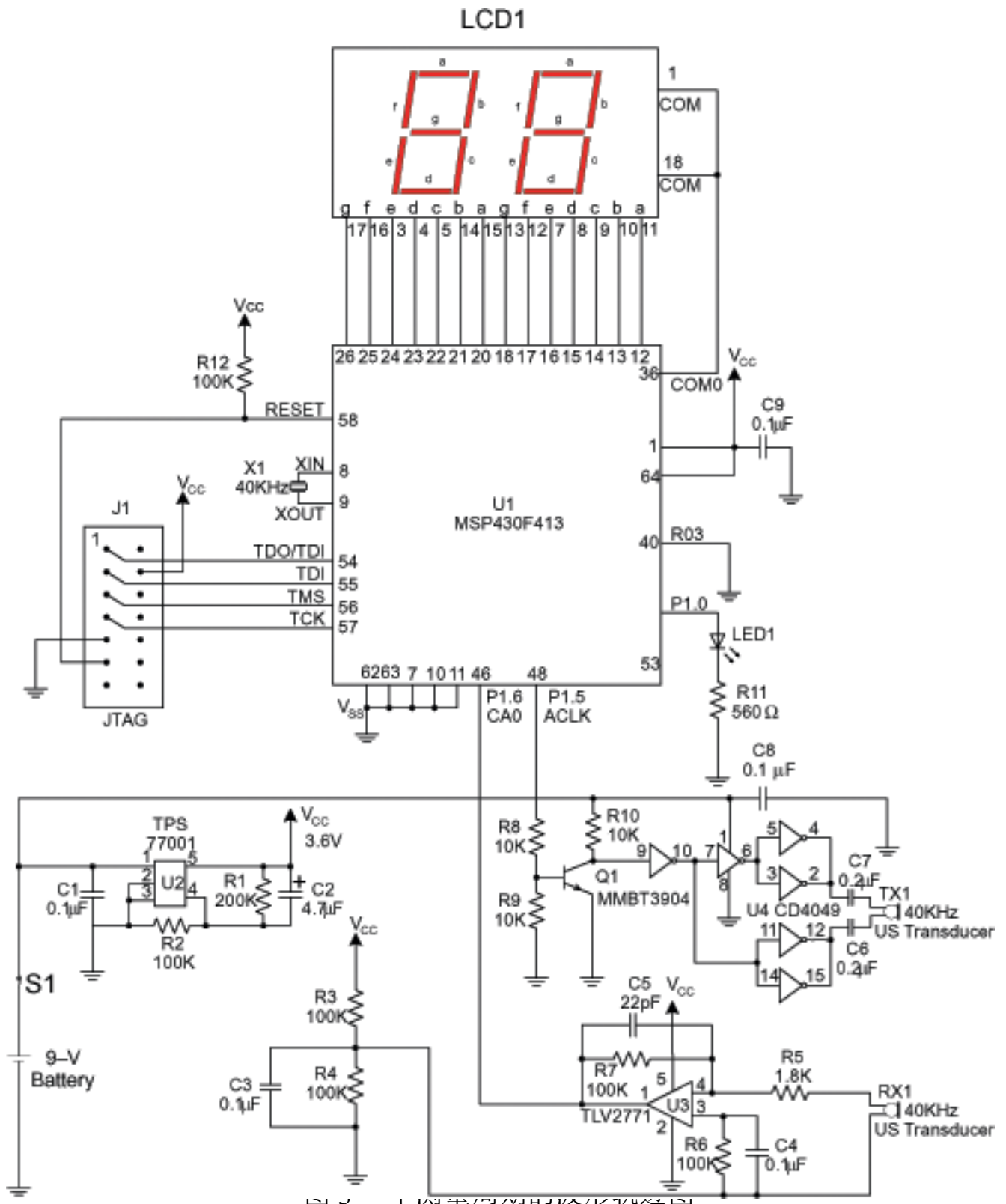


图 3-1-1 用单片机测时宽电路原理图



3 软件 Ultrasonic.s43

3.1 设备初始化子程序

这个子程序初始化及设置外围电路。首先关闭看门狗电路，采用一个软件延时让低频晶振稳定，将 FLL+倍频器设置为 64 来产生 2.56MHz 的 MCLK 频率。P1.0 设置为输出控制 LED。未使用的端口引脚设置为输出，端口引脚 P1.5 设置为输出带缓冲的 40KHz ACLK 频率。基本定时器 1 启动并设置为提供一个 150Hz LCD 频率并每 205 毫秒中断 CPU 开始一次测量周期。比较器 A 内部参考电平设置为 0.5V_{cc}，CAPD 位设置为关闭比较器输入引脚的输入缓冲。LCD 模块打开并在本应用中设置为静态模式驱动两位静态 LCD。LCD 存储器清零，LCD 的初始显示为 00。基本定时器 1 中断和全局中断打开，这样基本定时器 1 就可以周期性中断 CPU。

3.2 主循环程序

主循环用存储在 DIGITS 缓冲区内的值更新液晶显示器然后将 MSP430 置于 LPM3 睡眠模式。MSP430 维持在睡眠模式直至基本定时器 1 中断发生及其 BT_ISR 返回活动模式。这时就开始一个测量周期。定时器 A 设置为 16 位增计数模式，选择 ALCK 作为其时钟源。CCR1 设置为比较模式，初始值为 12 以便在 P1.5 引脚上输出 12 个周期的 40KHz 的脉冲序列。接下去是一个 36ACLK 周期的延时，保证输出传感器稳定。这是通过设置 CCR1 为初始值 36 的比较模式实现的。在 CCR1 处于比较等待状态时，MSP430 保持 LPM0 状态。

然后系统切换到通过接收传感器接收回声。比较器 A 被设置为等待回声，在回声到达的瞬间提供一个捕获中断。定时器 A 计数值从捕获一比较

寄存器 CCR1 中得到。这个值就是超声波序列从发射传感器到达目标然后返回这段距离所需的时间。计数值加上 48 加以调整，以补偿 12 个周期脉冲序列和 36 个周期发射传感器稳定延时的损失。CCR1 调整后的值就是脉冲序列发出时刻到回声到达系统这段时间的准确值。然后，就调用计算子程序计算实际距离（单位为英寸）并返回结果。如果结果超出范围，回声信号就接收不到，比较器 A 也不会产生捕获中断。MSP430 保持 LPM 状态知道下一个基本定时器 1 中断将其唤醒，通过检测其 CCTL1 控制寄存器中的 CAIFG 位来确认回声没有接收到。为了反映这种情况，在 DIGITS 中存储值 0XBE 在 LCD 上显示“E”。程序最终返回到主程序，更新 LCD 显示并返回 LPM3 睡眠模式。下一个基本定时器 1 中断将 MSP430 唤醒至活动状态重复程序执行过程。

3. 3 计算子程序

计算子程序负责本应用中的数学计算。CCR1 中调整后的 16 位数值存储为变量 Result，这个值就代表超声波序列从系统到目标然后返回系统这段距离所需的时间。因为定时器 A 计数间隔为 25 微秒，对应的时间值为 Result × 25 微秒。假设室温下声音的速度为 1100 英寸/秒，可以看出，从定时器 A 计数值得到的 Result 每 6 个值对应于 1 英寸的距离。

为了采用 MSP430 现有的整数计算达到要求的精度，16 位的 Result 首先乘以 100 然后除以 6。16 位 × 16 位的乘法是由子程序 Mu100 完成的，32 位的结果存储在变量 htX100_msw 和 htX100_lsw 中，然后除以 6 并将结果保存在变量 DIGITS 中。DIGITS 中的值是二进制格式的，由 hex2bcd 子程序将这个二进制值转换成 BCD 码，然后丢弃 BCD 码的最后两个数字来补

偿前面所作的与 100 的乘法。两位的结果值返回到变量 DIGITS 中。

3. 4BT_ISR 子程序

基本定时器 1 中断子程序 BT_ISR 对保存在堆栈中的状态寄存器 SR 中的位进行操作，以使 MSP430 能够返回活动状态，继续执行主循环中 LPM3 之后的程序代码。

3. 5 显示子程序

这个子程序用变量 DIGITS 的值更新两位静态 LCD 显示。静态显示器的段值存储在查寻表 LCD_Tab 中。通过将 DIGITS 中的数字与 LCD_Tab 查寻表中的位置相关联，LCD 存储器就可以装入所需要的段值。

3. 6 延时子程序

这个子程序产生一个 16 位的软件延时。由于软件减计数的变量处于堆栈的顶部 (TOS)，寄存器不会受到影响。延时时间到后，子程序返回前堆栈指针 (SP) 恢复到原始值。

4 结论

内部集成的模拟比较器 A、带硬件捕获/比较寄存器的 16 位定时器 A、基本定时器 1、LCD 驱动电路简化了超声波测距应用的设计，提供了一种单片系统解决方案。本应用在 15 英寸距离测量中的平均功耗为 1.3 毫安。这包括 LDO U2、运算放大器 U3 和 CMOS 二进制反相器 U4 的静态电流，运算放大器自身就由 1 毫安的静态电流，电路其他部分的功耗为 300 微安。LED 工作时需 5 毫安电流，MSP430 在 LCD 连续工作时需 2.1 微安的平均电流。这是利用 MSP430 的低功耗特性实现的，MSP430 大部分时间处于 LPM3 睡眠状态，这时用到的 CPU 资源仅为 5.6%。



因为声波的速度与温度有关，测量值在非室温下准确度会下降。在本应用中引入一个简单的基于热电阻的温度测量和距离补偿，可以使系统在很大的温度范围内精确测量。如果需要，测得的距离和温度数据可以存储在 FLASH 存储器中。加上额外增益和使用多态 LCD 来读出尽可能多的位数也可以增大测量范围。

5 参考文献

1. MSP430x41x 混合信号微控制器技术手册 SLA340
2. MSP430x4xx 系列用户手册，SLAU056
3. MSP430 系列混合信号微控制器，用户报告 SLAA024
4. TPS770xx 超低功耗 LDO 线性稳压器技术手册 SLVS210
5. TLV277x 系列高速运算放大器技术手册 SLOS209
6. CD4049UB,CMOS 二进制缓冲反相器技术手册 SCHS046A

附录 A 程序代码



; 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
; 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 is. You may,
; only under TI 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 nonTI 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

Ultrasonic Distance Measurement With the MSP430 10

mov.w #CCIE,&CCTL1 ; Compare mode,interrupt

bis.w #LPM0,SR ; Wait for CCR1 interrupt

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 ??

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+TACLK+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 #TACLRL,&TACTL

mov.w #36,&CCR1 ; Delay for transducer to settle

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 找?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 担

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 担

SetupP6 mov.b #000h,&P6OUT ; Clear P6 output register

bis.b #0ffh,&P6DIR ; Unused pins as o/p 担

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

Ultrasonic Distance Measurement With the MSP430 12

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

SLAA136A

; 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

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? 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 ??

db b+c ; Displays ??

db a+b+d+e+g ; Displays ??

db a+b+c+d+g ; Displays ??

db b+c+f+g ; Displays ??

db a+c+d+f+g ; Displays ??

db a+c+d+e+f+g ; Displays ??

db a+b+c ; Displays ??

db a+b+c+d+e+f+g ; Displays ??

db a+b+c+d+f+g ; Displays ??

db a+b+c+e+f+g ; Displays 撇?

db Blank ; Displays Blank

db a+d+e+f ; Displays 攔?

db b+c+d+e+g ; Displays 擠?d

db a+d+e+f+g ; Displays 擡?

db a+e+f+g ; Displays 擣?

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

郑重申明



德州仪器公司和它的分支机构保留在任何时候对其产品和服务进行更正、修改、增加、完善和其他变化以及终止产品和服务的权利，无须另外声明。用户应当在订货前获取最新消息并验证获得信息是完整和当前的。所有产品的销售以订单确认时 TI 的货期和销售状况为准。

TI 根据销售时说明书的标准授权对其硬件产品履行授权。TI 在认为有必要支持该授权的范围内采取测试和其他质量控制技术。除非政府委托要求，对每一产品所有参数进行测试是不必要的。

TI 没有责任提供应用帮助或者用户产品设计。用户使用 TI 元器件时自行对他们的产品和应用负责。为了降低用户产品和应用的风险，用户应提供足够的设计和操作系统安全措施。

TI 不对任何其产品或服务中使用到的与联合、机器或过程相关的知识产权，比如 TI 的专利、版权、源代码等，不论是明确的或是暗含的许可证进行授权。TI 印刷的关于第三方产品或服务的信息不代表 TI 对使用这些产品和服务的许可、授权或者签注。使用这些信息也许需要得到第三方的专利许可或第三方的其他知识产权许可，或者获得 TI 的专利和知识产权许可。

TI 技术手册和技术资料内信息的复制品仅在没有改变的情况下才被允许，而且不得对其中的附属的授权、条件、限制和声明进行删减。任何对复制品信息的变动都是不公正的和欺骗性的商业行为。TI 不会对这些改动后的文件负责。

传售 TI 产品或服务时，如果超出了 TI 公司对该产品和服务的参数的声明，就侵犯了与产品或服务相关的明确的或者暗含的授权，是一种不公正的欺骗性的商业行为。TI 不对此类声明负责。