Application Report
SLAA337–October 2006

Interfacing the DAC8832 to the MSP430F449

Jojo Parguian *Data Acquisition Products*

ABSTRACT

This application report describes how to interface the DAC8832, an ultra-low power, 16-bit resolution, voltage-output digital-to-analog converter, to the MSP430F449 Mixed Signal Microcontroller using the HPA449 from SoftBaugh (http://www.softbaugh.com).

Contents

1 Introduction 1
2 Hardware Setup Configuration 2
3 Principle of Operation 3
4 Generating the Sinewave Output 4
5 Summary 5
6 References 5
Appendix A MSP430F449 Software Code 6

List of Figures

1 HPA449 Hardware Configuration 2
2 DAC8832EVM Hardware Configuration 2
3 MSP430 and DAC8832 Circuit Diagram 3
4 DAC8832 Serial Interface Timing 3
5 Actual Timing of the DAC8832 SPI Serial Interface 4
6 DAC8832 Reference and Output Waveform 5

1 Introduction

The DAC8832 is a single channel, low-power, low-noise, low glitch, 16-bit resolution, fast-settling, voltage output digital-to-analog converter (DAC) that features a fast serial interface of up to 50MHz clock speed. The communication port of the DAC can accept 16-bits serial input data that will be interfaced with the MSP430F449 using the SPI™ protocol as described in this application report. The DAC power supply, V_{DD}, provides the power for both analog and digital sections and can range from a minimum of +2.7V to a maximum of +5.5V. An external reference voltage is required to set the range of the full-scale voltage of the DAC8832, which can be set from a minimum of 1.25V up to a maximum of V_{DD}.

The conventional way of using this particular DAC is to add an external operational amplifier on its output in order to prevent loading the DAC. The DAC8832 output is unbuffered, so the user can tailor the DAC output error as well as its power consumption by specifically selecting the operational amplifier to be used. The OPA277 was selected for the investigation discussed here; this op amp provides the basis of the DAC settling time, linearity and other operational factors. Proper selection of the external op amp is crucial to the overall design.

2 Hardware Setup Configuration

This application report describes an investigation using the HPA449 platform for the MSP430F449 and the DAC8832EVM. Once the HPA449 and the DAC8832EVM are configured properly, both devices can be connected together very easily. Figure 1 through Figure 3 illustrate the hardware setup for both the DAC8832EVM and the HPA449 boards.

The HPA449 comes pre-configured with the correct jumper settings from the factory; see Figure 1.

The hardware setup for the DAC8832EVM (shown in Figure 2) depicts the simple diagram of the interface connection between the DAC8832 and the MSP430F449, as seen in Figure 3.

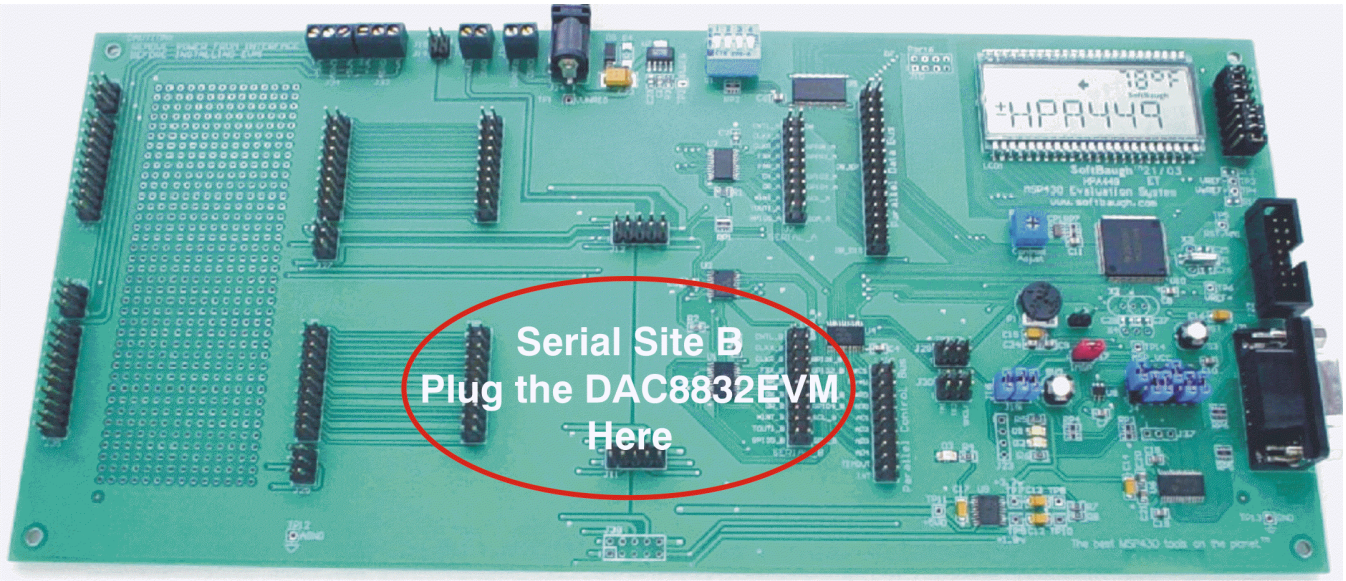
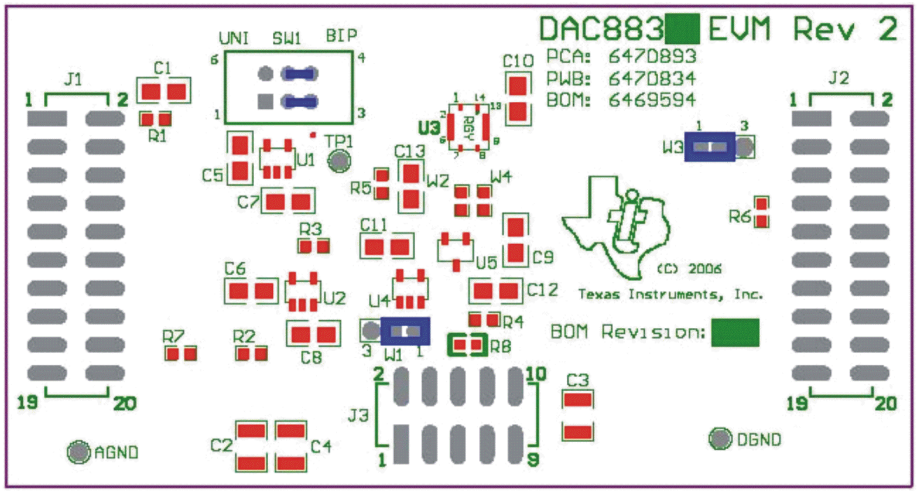


Figure 1. HPA449 Hardware Configuration



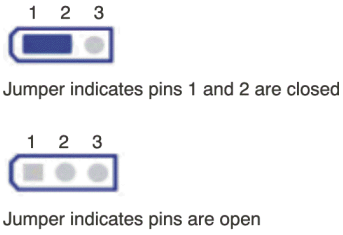


Figure 2. DAC8832EVM Hardware Configuration

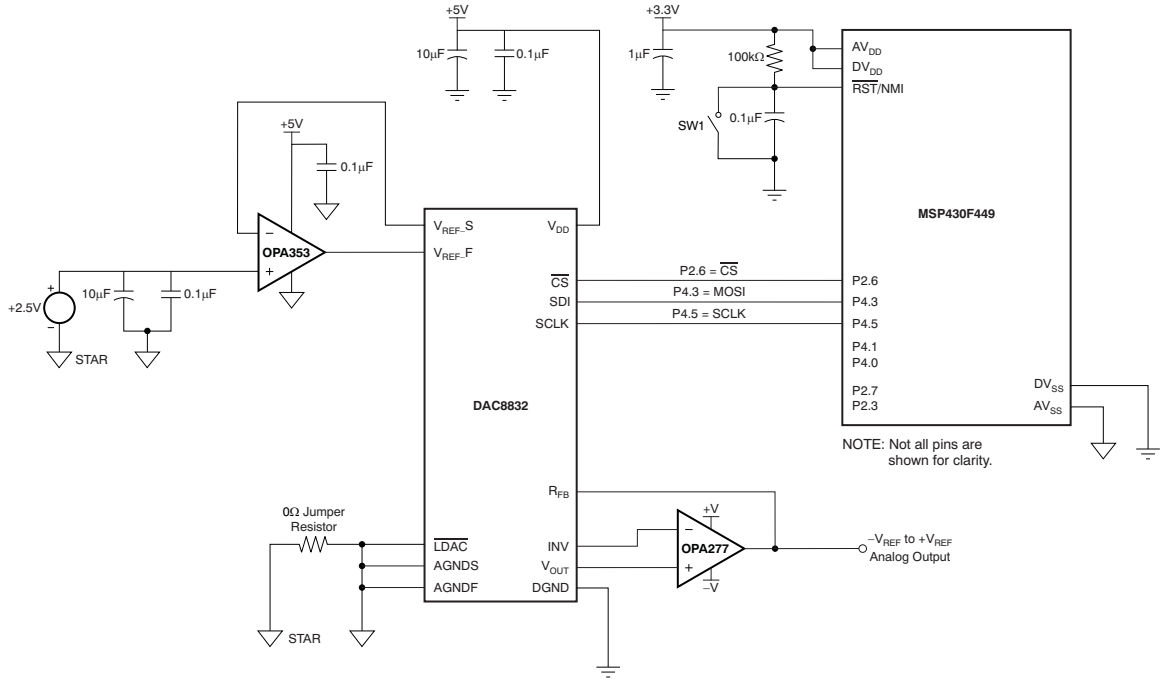


Figure 3. MSP430 and DAC8832 Circuit Diagram

3 Principle of Operation

The MSP430F449 microcontroller interfaces with the DAC8832 using the serial peripheral interface (SPI) serial data communication protocol via the MSP430 USART1 port. Only two pins of the four-pin SPI mode of configuration are used. The interface is implemented this way because there is no need for reading any data back from the DAC8832 or having the MSP430 microcontroller function as a slave (for SPI purposes) to another host peripheral. Therefore, the STE and the MISO functions in the SPI mode of the USART1 port are not utilized.

The $\overline{CS}$ function enables the SPI port and latches data into the DAC input register. This function is accomplished using a GPIO pin, P2.6, of the MSP430 microcontroller. When $\overline{CS}$ transitions from high to low, the serial input data bits are advanced one bit at a time on each rising edge of SCLK (on pin P4.5 of the MSP430 microcontroller) into the input shift register. The DAC8832 receives the last 16 bits of the digital input word serially while the $\overline{CS}$ pin is low. Since the SPI mode only provides eight data clocks per transmission, two write cycles are required within the $\overline{CS}$ low period; see Figure 4. The 16 bits of data are transferred into the shift register, MSB first, via the MOSI pin on P4.3 of the MSP430 device. However, if the $\overline{CS}$ pin is raised high before the 16th SCLK cycle is finished, the data becomes corrupted.

The DAC input register is updated or the data is latched following the low-to-high transition of the $\overline{CS}$ signal after the 16TH SCLK cycle. If more than 16 clocks are generated while $\overline{CS}$ is low, only the last 16 bits are transferred into the input register on the rising edge of $\overline{CS}$.

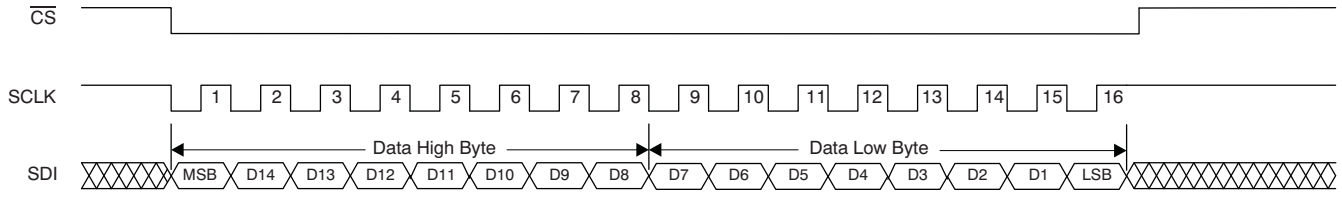


Figure 4. DAC8832 Serial Interface Timing

4 Generating the Sinewave Output

The software code associated with this application report (see Appendix A) is available from the launch page for this application report on the Texas Instruments web site as SLAA337.zip and contains the workspace and the associated project for the MSP430 KickStart™ version of the IAR Embedded Workbench™ from IAR Systems. Once the code is extracted to your PC, locate the DAC8832 file folder and open the associated workspace and project. If all your hardware is correctly set up, load the executable code by clicking on the ***Debug*** icon of the IAR Embedded Workbench IDE. Run the program by clicking on the ***Go*** icon in the debug window of the IAR Embedded Workbench IDE.

If the code executes properly, the actual timing diagram of the SPI serial interface will be achieved (see Figure 5). Channel 1 shows the $\overline{CS}$ signal with a period of approximately 7.13µs while channel 2 shows the SCLK running at approximately 4MHz. Channel 3 shows the MOSI line (serial data in) transmitting the 16-bit data word; channel 4 shows the corresponding DAC output voltage. The DAC8832 is operating in transparent mode because the $\overline{LDAC}$ pin is tied to ground. Thus, the DAC output is updated simultaneously with the content of the DAC input register following the low-to-high transition of the $\overline{CS}$ signal.

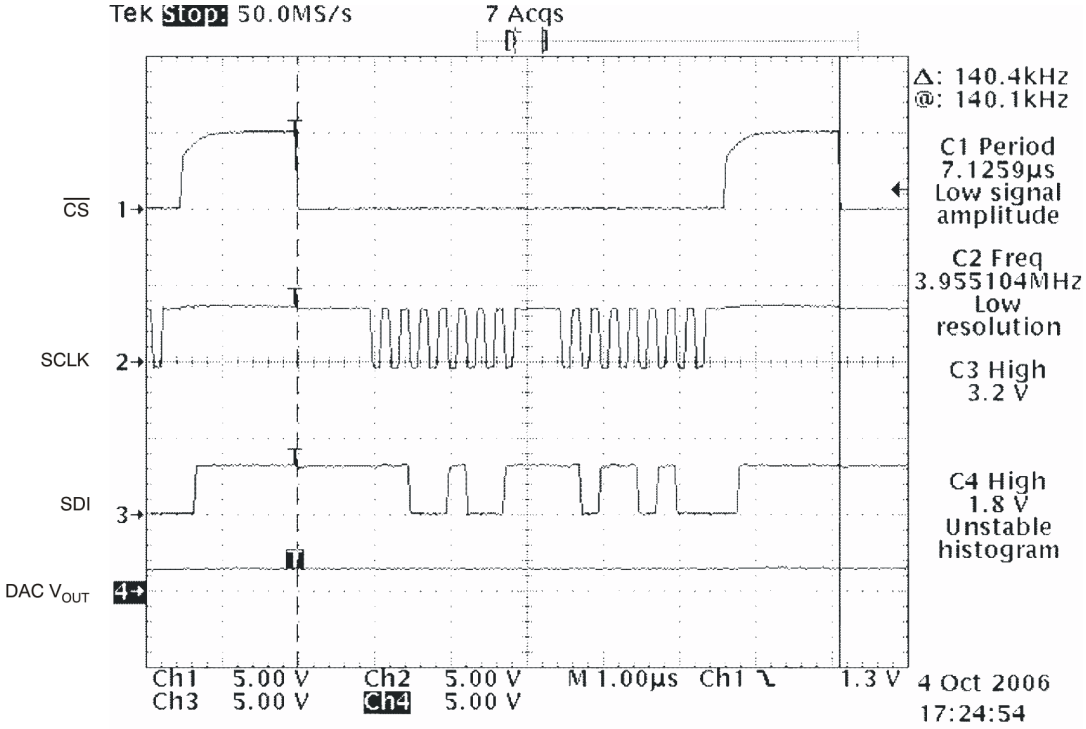


Figure 5. Actual Timing of the DAC8832 SPI Serial Interface

The code generates a sinusoidal waveform based on the 256 sample points provided as a lookup table in the program. Once everything is initialized and the SPI port is enabled, the code writes the values from the sine table into the DAC and then enters an endless loop until the program is aborted. The waveform below (Figure 6) is the resulting sinusoidal output with a frequency of approximately 550Hz based on the 7.13µs update rate.

Figure 6 shows the reference voltage, V_{REF}, set to +2.5V dc as seen on channel 1. On channel 2, the sinusoidal waveform output from –2.5V to +2.5V is displayed, because the DAC output is configured for the bipolar mode of operation as shown Figure 3.

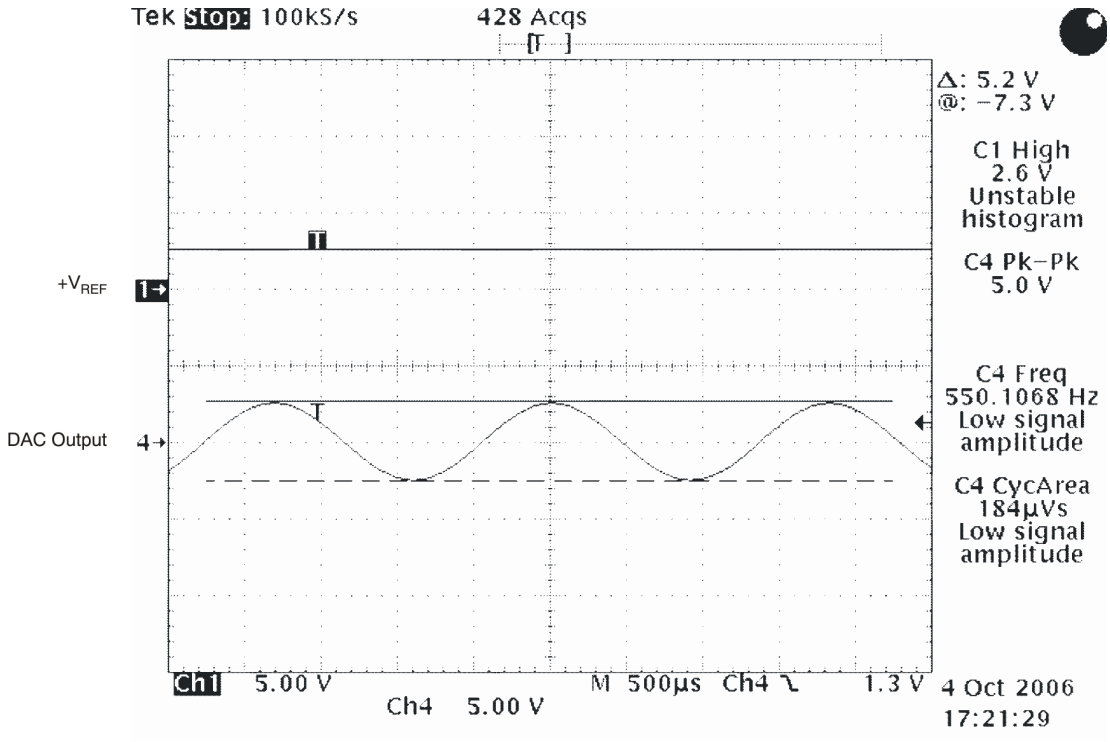


Figure 6. DAC8832 Reference and Output Waveform

5 Summary

This application note presents a straightforward method of interfacing the DAC8832 to the MSP430F449 microcontroller using the SPI mode of serial communication. Using the software program provided in this application note, a simple routine to generate a sinusoidal waveform (or other multiplying function) is achieved. Integrating the DAC8832EVM with the HPA449 evaluation system made this task even easier. Refer to the product data sheet for detailed information regarding the DAC8832. You can also contact TI's Data Acquisition Product group for further support by sending an e-mail to dataconvapps@list.ti.com.

For questions or information required regarding the HPA449 evaluation system, please contact SoftBaugh, Inc.

6 References

Except where noted, all documents referenced in this application report are available for download from the Texas Instruments web site at www.ti.com.

- DAC8832 Product Data Sheet (SBAS380)
- DAC8831/32EVM User's Guide (SLAU153)
- MSP430F449 Product Data Sheet (SLAS344)
- MSP430X4XX Family User's Manual (SLAU056)
- MSP430F44X Evaluation System (HPA449) User's Guide. Available from SoftBaugh, Inc.

Appendix A MSP430F449 Software Code

Main Code

```
;*******************************************************************************
; MSP430F449 Demo - SPI Communication with DAC8832 SPI function Using HPA449 v1.1
; Assembled with IAR Embedded Workbench for MSP430 Kickstart
; Author:  Jojo Parguian
;               HPA/DAP
; Company: Texas Instruments, Inc.
; Used:
;        HPA449 V1.1
;        DAC8832 EVM Rev 2
; Note: Use HPA449 Serial Port B
;*******************************************************************************
#include   "msp430x44x.h"         // Standard Equations
#include   "legal.asm"
#include   "readme.asm"
#define      DATASPI  R9
/*************************************************************
* Pin Assignment - GPIO Definitions
*************************************************************/
/* J2 Connections */
#define CSb              0x40  /* P2.6 */
#define FSb              0x04  /* P1.2 */
/*************************************************************
* Miscellaneous MSP430 Register Definitions
*************************************************************/
#define SPI              0x038

;-------------------------------------------------------------------------
; 16-bit Unipolar Sine Lookup table with 256 steps
;-------------------------------------------------------------------------
          ORG   01000h
;-------------------------------------------------------------------------------------------
Sin_tab   DW    32768,33572,34376,35178,35980,36779,37576,38370,39161,39947,40730,41507,42280
    DW    43046,43807,44561,45307,46047,46778,47500,48214,48919,49614,50298,50972,51636
    DW    52287,52927,53555,54171,54773,55362,55938,56499,57047,57579,58097,58600,59087
    DW    59558,60013,60451,60873,61278,61666,62036,62389,62724,63041,63339,63620,63881
    DW    64124,64348,64553,64739,64905,65053,65180,65289,65377,65446,65496,65525,65535
    DW    65525,65496,65446,65377,65289,65180,65053,64905,64739,64553,64348,64124,63881
    DW    63620,63339,63041,62724,62389,62036,61666,61278,60873,60451,60013,59558,59087
    DW    58600,58097,57579,57047,56499,55938,55362,54773,54171,53555,52927,52287,51636
    DW    50972,50298,49614,48919,48214,47500,46778,46047,45307,44561,43807,43046,42280
    DW    41507,40730,39947,39161,38370,37576,36779,35980,35178,34376,33572,32768,31964
    DW    31160,30358,29556,28757,27960,27166,26375,25589,24806,24029,23256,22490,21729
    DW    20975,20229,19489,18758,18036,17322,16617,15922,15238,14564,13900,13249,12609
    DW    11981,11365,10763,10174,9598,9037,8489,7957,7439,6936,6449,5978,5523,5085,4663
    DW    4258,3870,3500,3147,2812,2495,2197,1916,1655,1412,1188,983,797,631,483,356,247
    DW    159,90,40,11,1,11,40,90,159,247,356,483,631,797,983,1188,1412,1655,1916,2197
    DW    2495,2812,3147,3500,3870,4258,4663,5085,5523,5978,6449,6936,7439,7957,8489,9037
    DW    9598,10174,10763,11365,11981,12609,13249,13900,14564,15238,15922,16617,17322
    DW    18036,18758,19489,20229,20975,21729,22490,23256,24029,24806,25589,26375,27166
    DW    27960,28757,29556,30358,31160,31964,32768
;-------------------------------------------------------------------------------------------
            ORG     0F000h
;-------------------------------------------------------------------------
```

```
;-------------------------------------------------------------------------
;Program Code
;-------------------------------------------------------------------------
            RSEG CODE
;-------------------------------------------------------------------------
RESET       mov.w   #0A00h,SP                       ; Initialize stack-pointer
            call    #Init_Sys                       ; Initialize system
            clr.w   R6
            bic.b   #0FFh,&P1OUT
            bic.b   #CSb, &P2OUT
            bic.b   #FSb, &P1OUT
            bic.b   #38h, &P4OUT
Write_Data
            mov.w   #0FFh,R6                        ; 256 Sample Counter
            mov.w   #0,R5                           ; Table pointer
Again
            mov.w   Sin_tab(R5),DATASPI                 ;Sine Table Data
            swpb    DATASPI                         ; MSB first
            bic.b   #FSb,&P1OUT                     ; Assert CS_, which is routed to FS_ on
the board
            mov.b   DATASPI,&U1TXBUF                ; Transmit data
WaitXMT0    bit.b   #UTXIFG1, &IFG2                 ; TXBUF ready?
            jnc     WaitXMT0
            swpb    DATASPI                         ; LSB next
            mov.b   DATASPI,&U1TXBUF
WaitXMT1    bit.b   #UTXIFG1, &IFG2                 ; TXBUF ready?
            jnc     WaitXMT1
            mov.w   #1,R15                          ; Wait until all bits are
Delay       dec.w   R15                             ; transmitted
            jnz     Delay
            bis.b   #FSb, &P1OUT                    ; deassert CS_
            incd.w  R5                              ; increment table pointer
            sub.w   #1,R6                           ; decrement sample counter
            jnz     Again                           ; Do another sample
            jmp     Write_Data                      ; Another cycle

;------------------------------
; Clear TX Flag
;------------------------------
CLEAR0
           bit.b #UTXIFG0,&IFG1                       ; TXBUF ready?
           jnc CLEAR0                                 ; 1 = ready
           bic.b #UTXIFG0,&IFG1
           ret

;------------------------------
; Clear TX Flag
;------------------------------
CLEAR1
           bit.b #UTXIFG1,&IFG2                       ; TXBUF ready?
           jnc CLEAR1                                 ; 1 = ready
           bic.b #UTXIFG1,&IFG2
           ret

;-------------------------------------------------------------------------
Init_Sys;  Modules and Controls Registers set-up subroutine
;-------------------------------------------------------------------------
StopWDT   mov.w     #WDTPW+WDTHOLD,&WDTCTL              ; Stop Watchdog Timer

SetupFLL2   bis.b   #FN_4,&SCFI0                        ; x2 DCO, 8MHz nominal DCO
            bis.b   #DCOPLUS+XCAP14PF,&FLL_CTL0         ; DCO+, configure load caps
            mov.b   #121,&SCFQCTL                       ;(121+1) x 2 x 32768 = 7.99 Mhz

SetupPorts
; Port 1
          bis.b #FSb, &P1DIR
```

```
            bis.b #FSb, &P1OUT

;  Port 2
            bis.b #CSb, &P2DIR
            bis.b #CSb, &P2OUT

;  Port 4
            bis.b #SPI,&P4SEL                              ; P4.3,4,5 SPI option select
            bis.b #SPI,&P4DIR
            bis.b #SPI,&P4OUT
SetupSPI0
            bis.b #USPIE0,&ME1                              ; Enable SPI TX/RX
            mov.b #CHAR+SYNC+MM,&U0CTL                      ; 8-bit SPI Master
            bis.b #SSEL0+SSEL1+STC,&U0TCTL
            mov.b #02h,&U0BR0
            mov.b #00h,&U0BR1
            mov.b #00h,&U0MCTL
            bis.b #UTXIE0, &P1IE

SetupSPI1
            bis.b #USPIE1,&ME2                              ; Enable SPI TX/RX
            mov.b #CHAR+SYNC+MM+SWRST,&U1CTL                ; 8-bit SPI Master
            bis.b #CKPL+SSEL0+SSEL1+STC,&U1TCTL             ; 3-pin SPI mode, SMCLK
            mov.b #002h,&U1BR0                              ; CKPL+CKPH gives SCLK idle high and
data
            mov.b #000h,&U1BR1                              ; sampled on the falling edge of SCLK
            mov.b #000h,&U1MCTL                             ; CKPL gives SCLK idle high and data
            bis.b #USPIE1,&ME2                              ; sampled on the rising edge of SCLK
            bic.b #SWRST, &U1CTL                            ; CKPH gives SCLK idle low and data
                                                            ; sampled on the rising edge of SCLK
            ret
;-------------------------------------------------------------------------------
          COMMON  INTVEC              ; MSP430x44x Interrupt vectors
;-------------------------------------------------------------------------------
            ORG     RESET_VECTOR
RESET_VEC   DW      RESET               ; POR, ext. Reset, Watchdog
            END
```

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.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265