Application Report
SLAA261–September 2005

Interfacing the DAC7558 to the MSP430F449

Jojo Parguian *Data Acquisition Products*

ABSTRACT

This application report describes how to interface the DAC7558 digital-to-analog converter (DAC) to the MSP430F449 mixed signal microcontroller.

Contents

List of Figures

List of Tables

1 Introduction

The DAC7558 is an octal channel, low power, 12-bit resolution, voltage output DAC, which features low glitch, low crosstalk, 12-bit linear and monotonic with double-buffered serial interface. The double-buffered register architecture is implemented to allow for simultaneous update of all DACs while writing new data to each input registers. The communication port accepts 24 bits of serial input data, which is interfaced with the MSP430F449 using SPI protocol.

The DAC7558 can be powered from a single-supply source of +2.7 V minimum to +5.5 V maximum; this application report describes a +5-V power supply applied to VDD and +3.3 V applied to IOV_{DD}. A built-in POR (power-on reset) circuit is also integrated to ensure that all the DAC outputs are at a known state of either zero or midscale on power up. The state of the RSTSEL pin dictates whether the DAC outputs go to zero (i.e., RSTSEL = 0) or midscale (RSTSEL = 1). In addition, if RSTSEL is set to zero, the input data format that the DAC7558 accepts is unsigned binary (USB format). Otherwise, if RSTSEL is set to one, the input data format that the DAC7558 accepts is binary twos complement (BTC format).

The voltage source for the reference supply of the DAC7558 comes from the REF3140, which provides 4.096 V with an accuracy of 0.2% maximum and a drift of 15ppm/°C.

2 Hardware Setup Configuration

This application report is based on an experiment using the HPA449 platform for the MSP430F449 and the DAC7558EVM revision A. Once the HPA449 and the DAC7558EVM are configured properly, they can be connected together easily. The following figures show the hardware configuration setup for both the DAC7558EVM and the HPA449 boards.

The HPA449 comes configured with the correct jumper settings from the factory (see Figure 1).

The hardware setup configuration for the DAC7558EVM (see Figure 2) depicts the simple diagram of the interface connection between the DAC7558EVM and the MSP430F449 (see Figure 3).

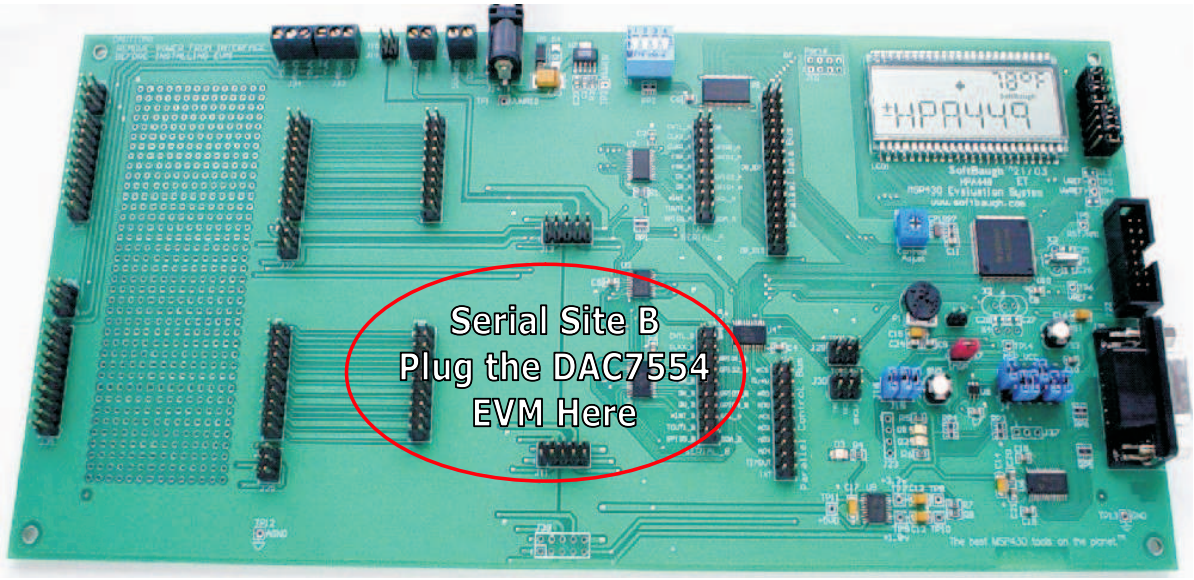

Figure 1. HPA449 Hardware Configuration

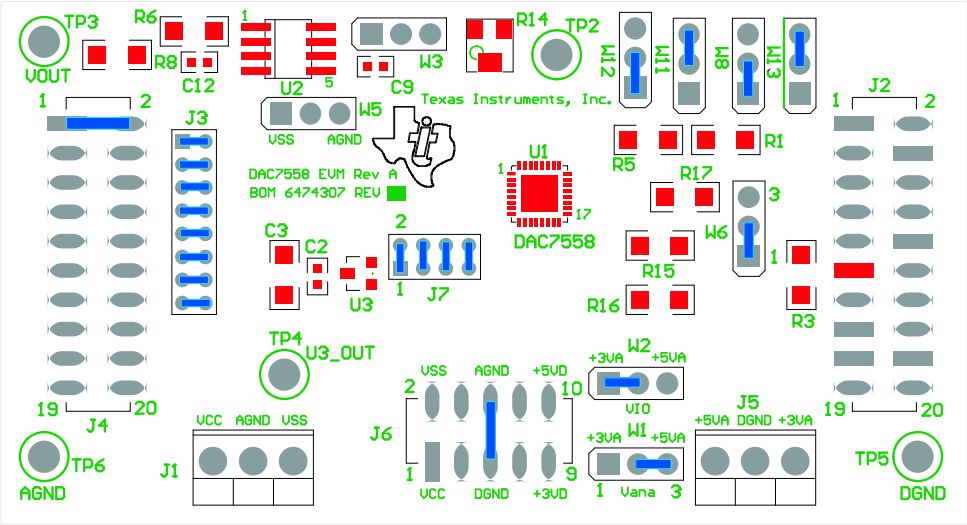

Figure 2. DAC7558 EVM Hardware Configuration

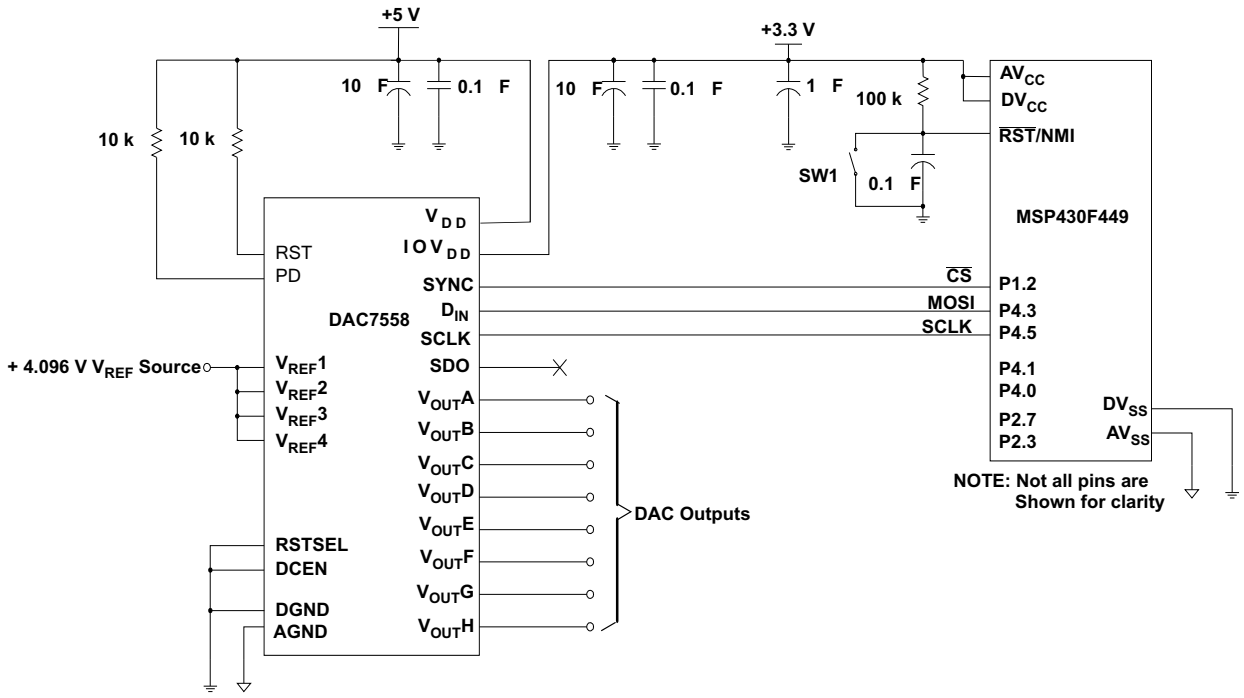

Figure 3. MSP430F449 and DAC7558 Circuit Diagram

3 Principle of Operation

The MSP430F449 microcontroller interfaces with the DAC7558 using the SPI serial data communication protocol via the MSP430 microcontroller's USART1 port. Only two pins are used out of the four-pin SPI mode of configuration. This is because it is unnecessary to read any data from the DAC7558, nor does the MSP430 microcontroller need to be slaved by another host peripheral for SPI purposes. Therefore, the STE and the MISO functions in SPI mode of the USART1 port are unused.

The SYNC function is an edge-triggered signal where the falling edge indicates the start of a serial data frame transfer through the SPI bus. This SYNC function is accomplished using a GPIO pin, P1.2, of the MSP430 microcontroller to enable the serial communication and data frame synchronization.

The DAC7558 receives a 24-bit digital input word serially. Because the SPI only provides eight data clocks per transmission, three write cycles are required within the SYNC low period to complete one write cycle to the DAC7558 (see Figure 4). Starting from the MSB, the first two bits (A1 and A0) must be zero for the DAC to respond. The DAC7558 does not respond to other combinations other then 00. The next two bits (LD1 and LD0) contain the load bits that select the type of load to be performed. The load bits work with the SEL2, SEL1, and SEL0 bits to select one of many different DAC load and update combinations. See Table 1 for the DAC7558 load and update combinations. The PWD bit is the power-down bit that enables the device through software to be powered down. The PWD bit must be zero for normal operation. Otherwise, if this bit is high, different power-down capability can be achieved depending on the different bit settings selected as illustrated in Tables 2 and 3 of the DAC7558 data sheet (SLAS435). The last 12 bits [MSB:LSB] compose the DAC7558 digital word with the most significant bit first. The last four bits are *don't care* bits and are ignored.

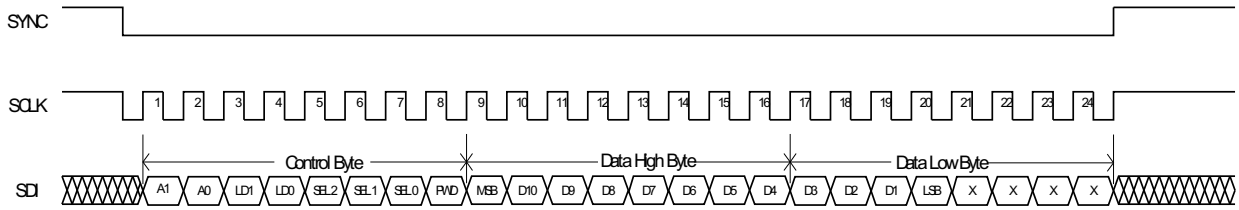

Figure 4. DAC7558 Serial Interface Timing

The falling edge of SCLK clocks the data in, starting from the MSB until all 24 bits are transferred into the shift register. Any data and clock pulses after the 24th falling edge of SCLK are ignored, and the transition of SYNC from low to high ends the data transfer. The data is automatically loaded from the shift register into the DAC input register at the end of the data transfer. If the SYNC signal is taken high before the 24th falling edge of SCLK, the data transfer is aborted, and the DAC input registers are not updated. As described previously, the control bits (LD1, LD0, SEL2, SEL1, and SEL0) are decoded by the DAC7558 and determine the type of load and update using the contents of the shift register.

The DAC7558 features a double-buffered architecture to allow new data to be written to each of the DAC registers without disturbing the analog outputs. The first sets of registers are the DAC input registers. The second sets of registers are the DAC output registers.

Table 1. DAC7558 Load and Update Combinations

DB23	DB22	DB21	DB20	DB19	DB18	DB17	DB16	DB15–DB4	DB3–DB0	DESCRIPTION
A1	A0	LD1	LD0	SEL2	SEL1	SEL0	PWD	MSB–LSB	Don't Care	
0	0	0	0	0	0	0	0	Data	X	Write to buffer A with data
(Both A1 and A0 should be set to zero for normal device operation. DAC(s) do not respond if any other combination is used)		0	0	0	0	1	0	Data	X	Write to buffer B with data
		0	0	0	1	0	0	Data	X	Write to buffer C with data
		0	0	0	1	1	0	Data	X	Write to buffer D with data
		0	0	1	0	0	0	Data	X	Write to buffer E with data
		0	0	1	0	1	0	Data	X	Write to buffer F with data
		0	0	1	1	0	0	Data	X	Write to buffer G with data
		0	0	1	1	1	0	Data	X	Write to buffer H with data
		0	1	(000, 001, 010, 011, 100, 101, 110, 111)			0	Data	X	Write to buffer with data and load DAC (selected by DB19, DB18, and DB17)
		1	0	(000, 001, 010, 011, 100, 101, 110, 111)			0	Data	X	Write to buffer with data and load DAC (selected by DB19, DB18, and DB17) and load all other DACs with buffer data
		1	1	0	0	0	0	Data	X	Load DACs A and B with current buffer data
		1	1	0	0	1	0	Data	X	Load DACs A, B, C, and D with current buffer data
		1	1	0	1	0	0	Data	X	Load DACs A, B, C, D, E, and F with current buffer data
		1	1	0	1	1	0	Data	X	Load DACs A, B, C, D, E, F, G, and H with current buffer data
		1	1	1	0	0	0	Data	X	Write to buffer with new data and load DACs A and B simultaneously
		1	1	1	0	1	0	Data	X	Write to buffer with new data and load DACs A, B, C, and D simultaneously
		1	1	1	1	0	0	Data	X	Write to buffer with new data and load DACs A, B, C, D, E, and F simultaneously
		1	1	1	1	1	0	Data	X	Write to buffer with new data and load DACs A, B, C, D, E, F, G, and H simultaneously
		X	0	(000, 001, 010, 011, 100, 101, 110, 111)			1	See Table 2 of the data sheet	X	Write to buffer and load DAC with Power-Down command to individual channel (selected by DB19, DB18, and DB17)
		X	1	(000, 001, 010, 011, 100, 101, 110, 111)			1	See Table 2 and Table 3 of the data sheet	X	Write to buffer and load DACs with Power-Down command to multiple channels (selected by DB19, DB18, and DB17)

4 Generating the Sine-Wave Output

The actual timing diagram of the SPI serial interface is shown in Figure 5. Channel 2 shows the SCLK running at approximately 4 MHz while channel 3 shows the SDI transmitting the 24 bits of control and data word. The control bits (A1, A0, LD1, LD0, SEL2, SEL1, SEL0, and PWD) are set to 0x3E so that the data in the shift register is loaded in the input registers and DAC output registers of all eight DACs, as well as updating all the DAC outputs ($V_{OUT}A$, $V_{OUT}B$, $V_{OUT}C$, $V_{OUT}D$, $V_{OUT}E$, $V_{OUT}F$, $V_{OUT}G$, and $V_{OUT}H$). Channel 1 is the SYNC signal that enables the serial communication interface of the DAC7558 and signals the start of data frame transmission.

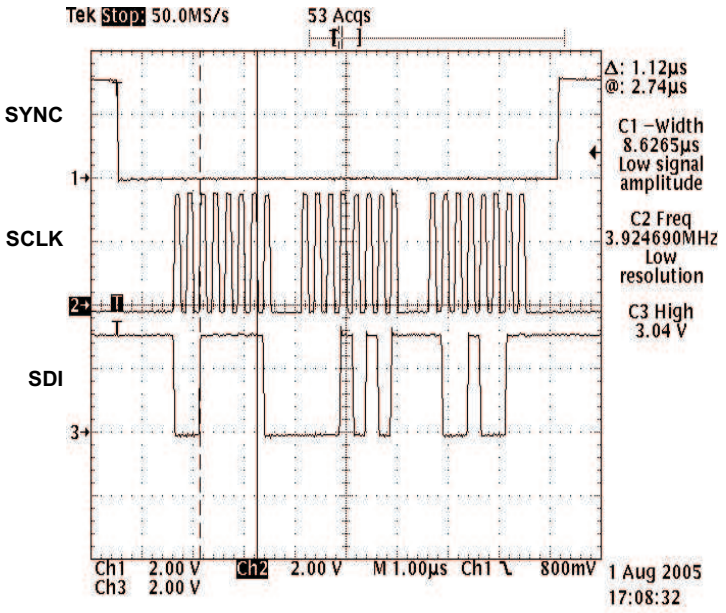

Figure 5. Actual Timing of the DAC7558 SPI Serial Interface

If the serial interface timing for the DAC7558 is met as shown in the Figure 5, the following sinusoidal waveform in Figure 6 should be observed. The DAC channel A output displays the sine wave with an amplitude of 8 Vpp while the rest of the DAC output channels are 4 Vpp. The signal amplitude of output A is 8 Vpp because the DAC A channel output of the DAC7558 is connected to an external output amplifier with a gain of 2 as shown in Figure 7. Only one DAC output channel at a time can be connected to the external amplifier and evaluated using the DAC7558EVM board. The illustration of Figure 6 only shows four selected DAC outputs because the oscilloscope used has only four probe channels. The rest of the DAC outputs display the 4-Vpp sinusoidal waveform, because the control word issued to the DAC7558 is to write to the DAC buffer with new data and load all DACs simultaneously.

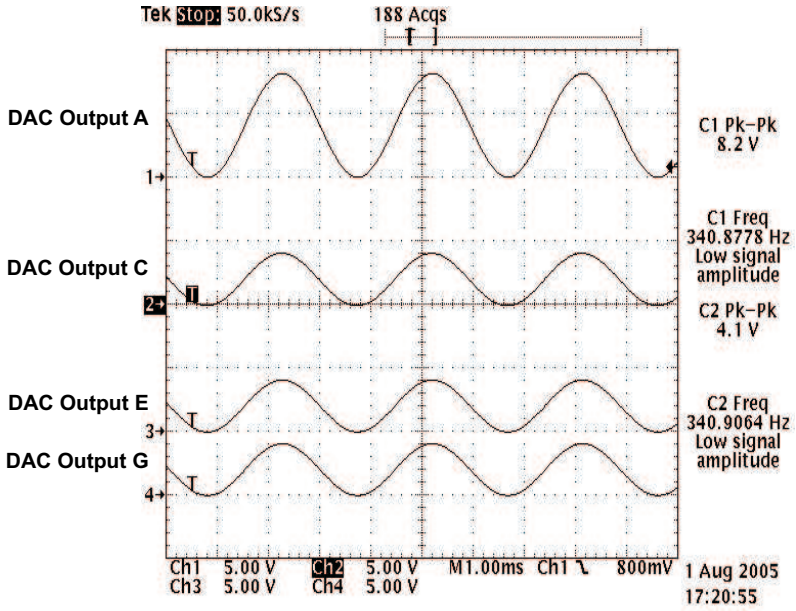

Figure 6. DAC Output Waveform Diagram

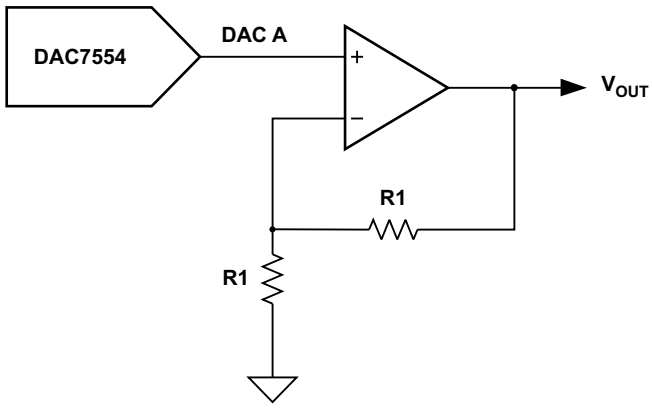

Figure 7. DAC A Output With a Gain of 2 (1 of 8 DACs)

5 Summary

This application report shows how easy it is to interface the DAC7558 to the MSP430F449 microcontroller using the SPI mode of serial communication. Using the software program provided in this application report, a simple routine to generate a sinusoidal waveform is achieved. Using the DAC7558EVM along with the HPA449 evaluation system makes it even easier. For more detailed information regarding the DAC7558, see data sheet SLAS435. To contact TI's Data Acquisition Product group for further support, send an e-mail to dataconvapps@list.ti.com.

For questions or information required regarding the HPA449 evaluation system, contact SoftBaugh, Inc. at e-mail address info@softbaugh.com, or call the company directly at the toll-free number (800) 794-5756 or commercial (770) 772-8111.

6 References

1. *DAC7558 12-Bit, Octal, Ultralow Glitch, Voltage Output, Digital-to-Analog Converter data sheet* (SLAS435)
2. *DAC7558 EVM User's Guide* (SLAU162)
3. *MSP430x43x, MSP430x44x, Mixed Signal Microcontroller data sheet* (SLAS344)
4. *MSP430x4xx Family User's Guide* (SLAU056)
5. MSP430F44x Evaluation System (HPA449) User's Guide (SoftBaugh, Inc)

Appendix A MSP430F449 Software Code

A.1 Main Code

```
;*******************************************************************
; MSP430F449 Demo - SPI Communication with DAC7558 and MSP430F449
;
; Assembled with IAR Embedded Workshop for MSP430 Kickstart
;
; Author:  Jojo Parguian
;          HPA/DAP
; Company: Texas Instruments, Inc.
;
; Used:
;          HPA449 V1.1
;          DAC7558 EVM Rev 1 & Rev A
;*******************************************************************
#include   "msp430x44x.h"        // Standard Equations
#include   "DAC7558_Test.h"     // DAC Equations
#include   "legal.asm"
#include   "readme.asm"

#define     DATASPI  R9
#define     Cmd_Word     0x3E   // Load all DACs with buffer data simultaneously and ALL DACs
update

;-------------------------------------------------------------------------
; 16-bit 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
```

```
Write_Data
            mov.w   #0FFh,R6
            mov.w   #0,R5

Again
            mov.w   Sin_tab(R5),DATASPI
            swpb    DATASPI                 ; MSB first
            bic.b   #F_SYNC, &P1OUT
            mov.b   #Cmd_Word,&U1TXBUF      ; Send command word out first
WaitXMT0
            bit.b   #UTXIFG1, &IFG2         ; TXBUF ready?
            jnc     WaitXMT0
            mov.b   DATASPI,&U1TXBUF        ; Send MSB now
            swpb    DATASPI                 ; LSB next
WaitXMT1
            bit.b   #UTXIFG1, &IFG2         ; TXBUF ready?
            jnc     WaitXMT1
            mov.b   DATASPI,&U1TXBUF        ; Send LSB now
WaitXMT2
            bit.b   #UTXIFG1, &IFG2         ; TXBUF ready?
            jnc     WaitXMT2
            incd.w  R5
            sub.w   #1,R6
            mov.w   #02h, R14
Delay0
            dec.w   R14                     ;
            jnz     Delay0                  ;
            bis.b   #F_SYNC, &P1OUT
            mov.w   #03h, R14               ; Allow some settling time
Delay1
            dec.w   R14                     ;
            jnz     Delay1
            and.w   #0FFh,R6
            jnz     Again
            jmp     Write_Data
;-------------------------------------------------------------------------
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 #F_SYNC, &P1DIR
          bis.b #F_SYNC, &P1OUT
; Port 2
          bis.b #CSb, &P2DIR
          bis.b #CSb, &P2OUT
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 #SSEL0+SSEL1+STC,&U1TCTL          ; 3-pin SPI mode, SMCLK
          mov.b #002h,&U1BR0
          mov.b #000h,&U1BR1
          mov.b #000h,&U1MCTL
          bis.b #USPIE1,&ME2
          bic.b #SWRST, &U1CTL
          ret

;-------------------------------------------------------------------------------------------
          COMMON  INTVEC              ; MSP430F449 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
		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