Application Report
SLAA297–March 2006

Interfacing the DAC8551 to the MSP430F449

Jojo Parguian Data Acquisition Products

ABSTRACT

This application report describes how to interface the DAC8551 digital-to-analog converter to the MSP430F449 mixed signal microcontroller.

Contents

List of Figures

List of Tables

1 Introduction

The DAC8551 is a single channel, low power, 16-bit resolution, voltage output DAC, which features ultra-low glitch, 16-bit linear and monotonic output with double buffered serial interface. The double buffered register architecture is implemented to allow for simultaneous update of the DAC output while writing new data to the input register. The DAC's communication port accepts 24 bits of serial data input, and is interfaced with the MSP430F449 using the SPI protocol for this report.

The DAC8551 can be powered from a single supply source of +2.7 V minimum to +5.5 V maximum. This application report shows a +5-V power supply applied to V_{DD} and is complaint with the logic voltage of +3.3 V from the microprocessor. A built-in POR (power-on reset) circuit is also integrated to ensure that the DAC output is at a known state (reset to zero) upon power up.

The voltage source for the reference supply of the DAC8551 comes from the REF02 precision voltage reference, which sets the DAC's output range to +5 V. REF02 has voltage output accuracy of ±0.2% max and a drift of 10 ppm/°C.

2 Hardware Setup Configuration

This application report is based on the HPA449 platform for the MSP430F449 and the DAC8551 EVM revision A. Once the HPA449 and the DAC8551 EVM are configured properly, they can be connected together very easily. The following figures show the hardware configuration setup for both the DAC8551 EVM and the HPA449 boards.

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

The hardware setup configuration for the DAC8551 EVM (shown in Figure 2) depicts the simple diagram of the interface connection between the DAC8551 and the MSP430F449, as shown in Figure 3.

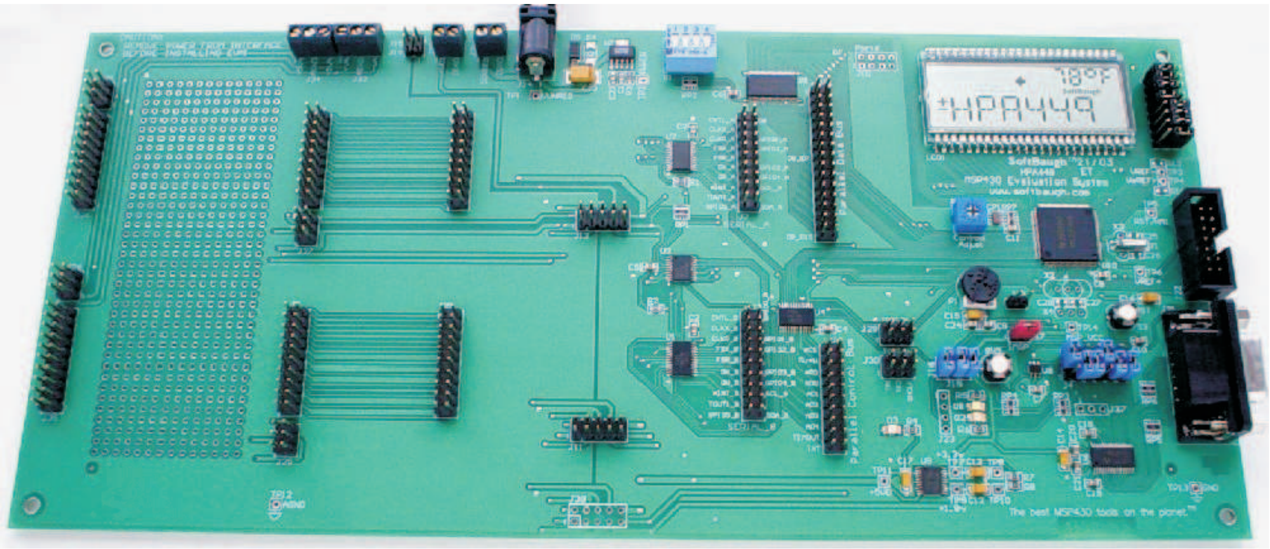

Figure 1. HPA449 Hardware Configuration

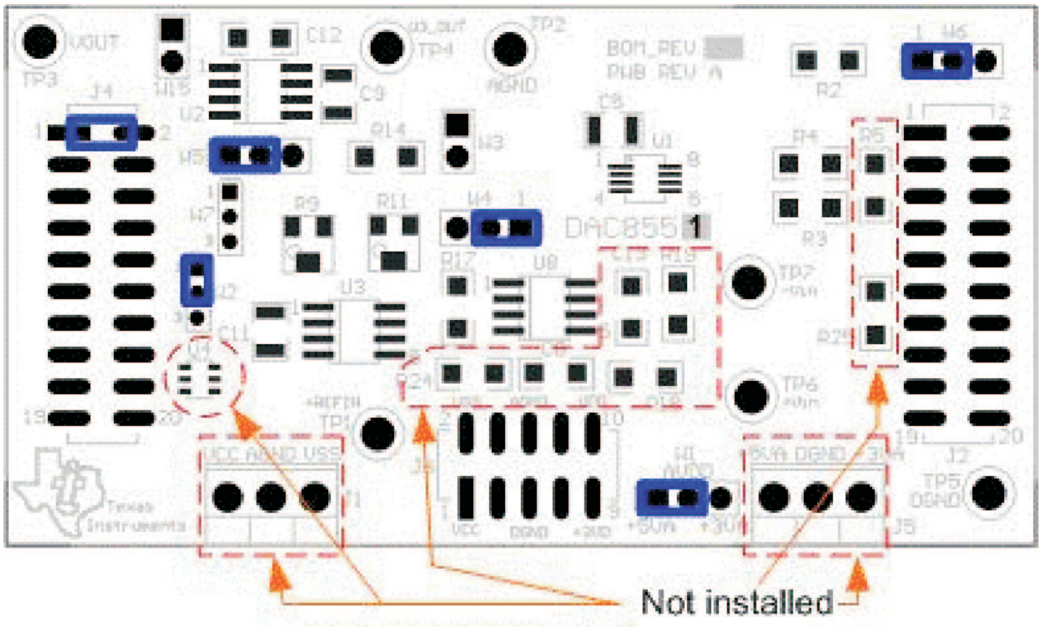

Figure 2. DAC8551 EVM Hardware Configuration

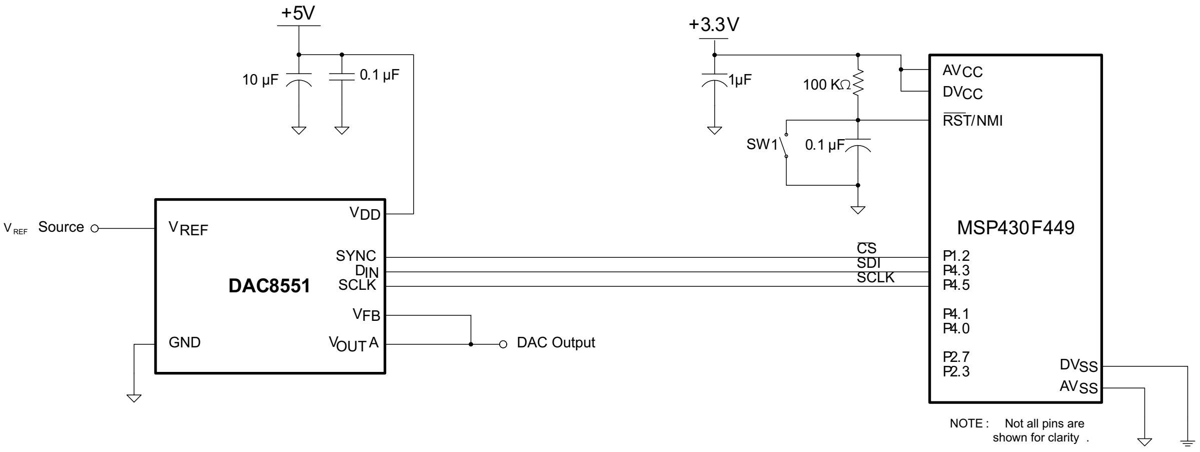

Figure 3. MSP530F449 and DAC8551 Circuit Diagram

3 Principle of Operation

The MSP430F449 µC interfaces with the DAC8551 using the SPI serial data communication protocol via the MSP430 µC's USART1 port. Only two pins are used out of the four-pin SPI mode of configuration. It is implemented this way as there is no need for reading any data back from the DAC8551 or having the MSP430 µC be slaved by another host peripheral for SPI purposes. Therefore, the STE and the MISO functions in SPI mode of the USART1 port are not utilized.

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

The DAC8551 receives a 24-bit digital input word serially. Since 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 DAC8551 as shown in Figure 4. The first six bits [DB23:DB18], starting from the MSB, are *don't cares* and are ignored. The next two bits (PD1 and PD0) are the control bits that set the mode of operation of the DAC8551 (normal mode or any of the three power-down modes). Refer to Table 1 for the device's different modes of operation. The last sixteen bits [DB15:DB0] compose the DAC8551 digital word with the most significant bit first.

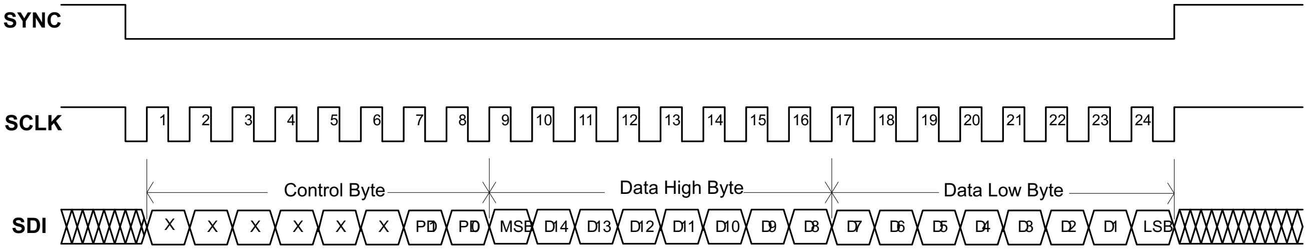

Figure 4. DAC8551 Serial Interface Timing

During a normal write sequence, the SYNC line should be kept low for at least 24 falling edges of SCLK. The falling edges of SCLK clock the data in starting from the MSB until all 24 bits are transferred into the shift register. Any data and clock pulses after the twenty-fourth falling edge of SCLK are ignored and the transition of SYNC from low to high ends the data transfer. The DAC register is updated on the 24th falling edge of SCLK. However, if the SYNC line is brought high before the 24th falling edge of SCLK, the shift register is reset and the data transfer is aborted. No DAC register update or change in operating mode occurs either.

As mentioned earlier, the DAC8551 features a double-buffered architecture to allow new data to be written to the DAC register without disturbing the analog output. The first set of the register is the DAC input register and the second set of the register is the DAC output register.

Table 1. DAC8551 Modes of Operation

PD1 (DB17)	PD0 (DB16)	OPERATING MODE
0	0	Normal operation
–	–	Power-down modes
0	1	Output typically 1 kΩ to GND
1	0	Output typically 100 kΩ to GND
1	1	High-Z

4 Generating the Sinewave Output

The actual timing diagram of the SPI serial interface is shown in Figure 5. Channel 1 shows the SYNC signal that enables the serial communication interface of the DAC8551 and signals the start of data frame transmission. Channel 2 shows the SCLK running at approximately 4 MHz while channel 3 shows the SDI transmitting the 24-bit control and data word. The control bits (PD1 and PD0) are set to 0x0 so that the device operates in normal mode. The 16-bit data shown is a randomly selected sample from the 256 samples in the sine table (refer to the software code in Chapter A).

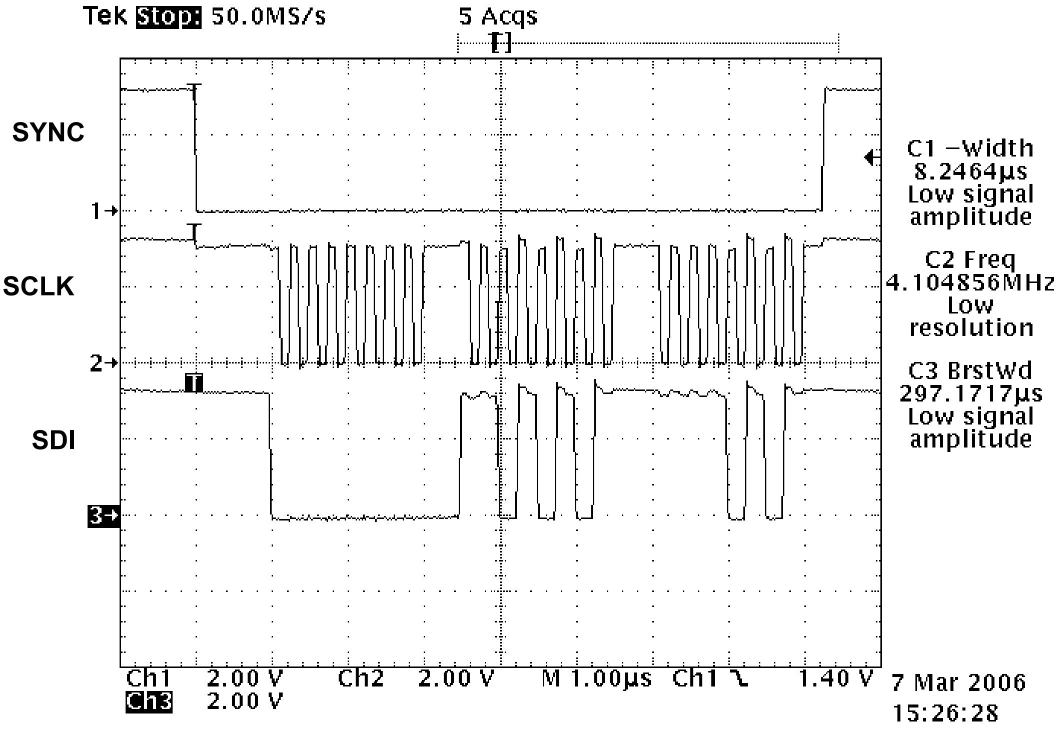

Figure 5. Actual Timing of the DAC8551 SPI Serial Interface

If the serial interface timing for the DAC8551 is met as shown in Figure 5, the sinusoidal waveform in Figure 6 should be observed. The DAC channel output displays the sinewave with amplitude of 5 V_{pp}.

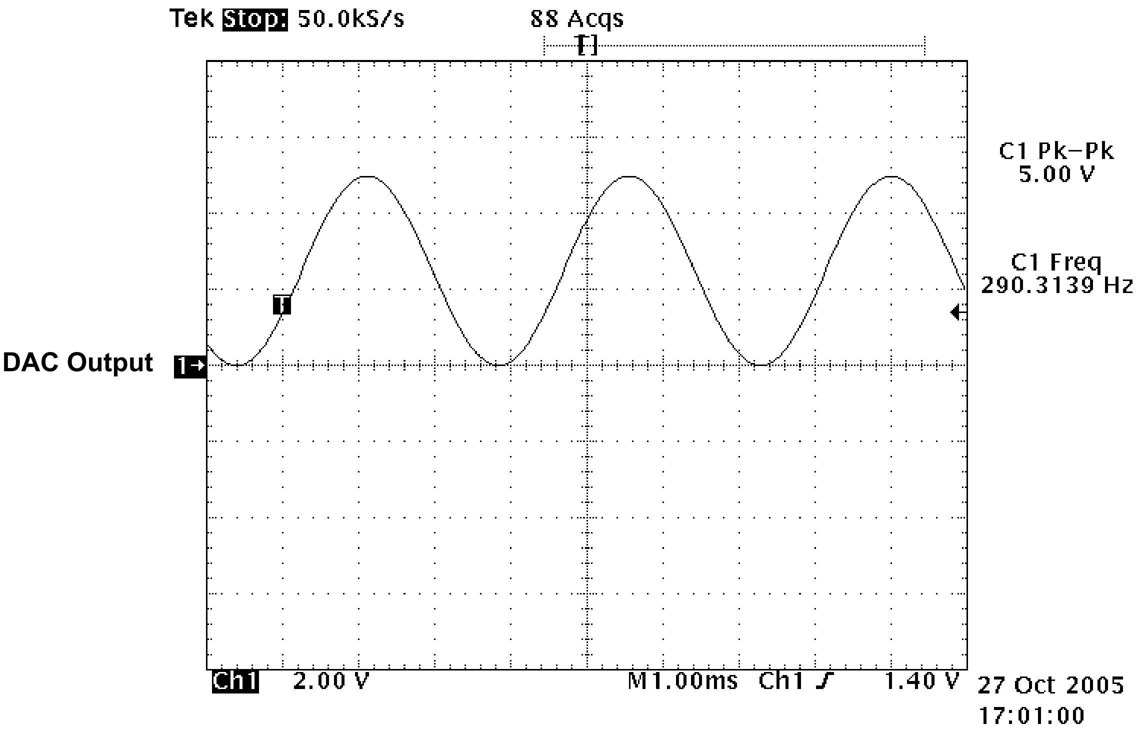

Figure 6. DAC Output Waveform Diagram

5 Summary

This application report shows how easy it is to interface the DAC8551 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. Utilizing the DAC8551 EVM along with the HPA449 evaluation system made it even easier. For more detailed information regarding the DAC8551, refer to the data sheet, SLAS429. For further support contact TI's Data Acquisition Product group by sending an e-mail to dataconvapps@list.ti.com.

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

6 References

1. *DAC8551 16-bit, Voltage Output, Serial Input DAC Data sheet* (SLAS429)
2. *DAC8551 EVM User's Guide* (SLAU172)
3. *MSP430F449 Datasheet* (SLAS344C)
4. *MSP430X4XX Family User's Guide Manual* (SLAU056C)
5. *MSP430F44X Evaluation System (HPA449) User's Guide* (SoftBaugh, Inc)

Appendix A MSP430F449 Software Code

A.1 Main Code

```
;*************************************************************************************************
; MSP430F449 Demo - SPI Communication with DAC8551 SPI function Using HPA449 v1.1
;
; Assembled with IAR Embedded Workbench for MSP430 Kickstart
;
; Author:  Jojo Parguian
;          HPA/DAP
; Company: Texas Instruments Incorporated
;
; Used:
;        HPA449 V1.1
;        DAC8551 EVM Rev 1 & Rev A
;*************************************************************************************************
#include   "msp430x44x.h"         // Standard Equations
#include   "legal.asm"
#include   "readme.asm"
#define      DATASPI    R9
#define      CSb        0x40        /* P2.6 */
#define      SPIb       0x028
;----------------------------------------------------------------------------
; 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

;************************************************************************
;Program Code
;************************************************************************
           RSEG CODE
;************************************************************************
RESET       mov.w   #0A00h,SP          ; Initialize stack-pointer
            call    #Init_Sys                 ; Initialize system
            clr.w   R6
            bis.b   #02h,&P3OUT
            bic.b   #0FFh,&P1OUT
Write_Data
            mov.w   #0FFh,R6
            mov.w   #0,R5
Again
```

```
            mov.w   #0h,DATASPI              ; Powerdown command (0 = normal operation)
            bic.b   #CSb, &P2OUT
            mov.b   DATASPI,&U1TXBUF
WaitXMTa0   bit.b   #UTXIFG1, &IFG2          ; TXBUF0 ready?
            jnc     WaitXMTa0
            mov.w   Sin_tab(R5), DATASPI     ;
            swpb    DATASPI                  ; MSB first
            mov.b   DATASPI,&U1TXBUF
WaitXMTa1   bit.b   #UTXIFG1, &IFG2          ; TXBUF0 ready?
            jnc     WaitXMTa1
            swpb    DATASPI                  ; LSB next
            mov.b   DATASPI,&U1TXBUF
WaitXMTa2   bit.b   #UTXIFG1, &IFG2          ; TXBUF0 ready?
            jnc     WaitXMTa2
            incd.w  R5
            sub.w   #1,R6
            mov.w   #02h, R14
Delay0      dec.w   R14                      ;
            jnz     Delay0
            bis.b   #CSb, &P2OUT
            and.w   #0FFh,R6
            jnz     Again
            jmp     Write_Data
;************************************************************************
Init_Sys;  Modules and Controls Registers set-up subroutine
;************************************************************************
StopWDT  mov.v   #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 2
          bis.b #CSb, &P2DIR
          bis.b #CSb, &P2OUT
; Port 4
          bis.b #SPIb,&P4SE                          ; P4.3,4,5 SPI option select
SetupSPI0
          bis.b #USPIE0,&ME1                           ; Enable SPI TX/RX
          mov.b #CHAR+SYNC+MM+SWRST,&U0CTL             ; 8-bit SPI Master
          bis.b #CKPL+CKPH+SSEL0+SSEL1+STC,&U0TCTL
          mov.b #002h,&U0BR0
          mov.b #000h,&U0BR1
          mov.b #000h,&U0MCTL
          bis.b #USPIE0,&ME1
          bic.b #SWRST, &U0CTL
SetupSPI1
          bis.b #USPIE1,&ME2                        ; Enable SPI TX/RX
          mov.b #CHAR+SYNC+MM+SWRST,&U1CTL          ; 8-bit SPI Master
          bis.b #CKPL+CKPH+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

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