

Interfacing the TLC3544/48 ADC to the MSP430F149

Tom Hendrick
Data Acquisition Products - Applications

ABSTRACT

The TLC3544 and TLC3548 14-bit serial output analog-to-digital converters can easily interface to the serial peripheral interface port of many popular microcontrollers. Using the serial port of the Texas Instruments MSP430F149, this application note focuses on configuring, sampling and converting analog data presented to the ADC. The assembly code developed for this application note shows how the EOC/ $\overline{\text{INT}}$ pin can be used as an interrupt source to the host processor.

Contents

1	Introduction.....	2
2	TLC3544/48 Evaluation Module.....	2
3	Serial Interface	3
3.1	Chip Select ($\overline{\text{CS}}$).....	3
3.2	Serial Data Input (SDI)	4
3.3	Serial Data Output Pin (SDO)	4
3.4	Serial Clock Pin (SCLK)	4
4	Control and I/O Pins	4
4.1	Conversion Start (CSTART)	4
4.2	Frame Sync (FS)	4
4.3	End of Conversion/Interrupt (EOC/ $\overline{\text{INT}}$)	5
4.4	Device Pin-Out	5
5	ADC Initialization and Operation	6
5.1	Initializing the ADC	6
5.2	Operating the ADC	6
5.3	EOC or $\overline{\text{INT}}$	7
6	MSPF149 Code Example	8

Figures

1	System Configuration	2
2	ADC Initialization Cycle	6
3	Sample/Convert Cycle.....	7
4	EOC/INT Timing	7

Tables

1	EVM to Micro Cable Definition.....	3
2	I/O Pins of the TLC3544/48.....	5

1 Introduction

The TLC3544 and TLC3548 are four- and eight-channel, 14-bit, serial ADCs. Each device has chip select (\overline{CS}), input clock (SCLK), serial data input (SDI), and serial data output (SDO) pins that can interface directly to a microprocessor. These data converters also feature a programmable end-of-conversion/interrupt (EOC \overline{INT}) pin, which can be used to initiate an interrupt service routine on the host processor.

This application note takes advantage of the hardware built into the TLC3544/48 evaluation module. A 20-pin ribbon cable from J12 on the data converter EVM to the microprocessor board is used to carry the necessary interface signals. The EVM's on-board signal generator provides the analog signals, and W17 allows the user to select either 3.3 V or 5.0 V as the digital I/O voltage. *Because the MSP430F149 is a low-voltage device, the EVM's I/O voltage must be set to 3.3 V before attempting to use this processor.*

Ten-pin, dual-row headers were added to the microprocessor boards to accommodate the cable from the data converter EVM. Figure 1 shows the system configuration, and Table 1 shows the connector pinout used.

The sample code presented in section 6 of this report was developed using the Texas Instruments MSP430F149 evaluation board. This simple code example demonstrates how to configure the serial port, initialize the data converter, and process an interrupt (via the EOC \overline{INT} pin) from the data converter.

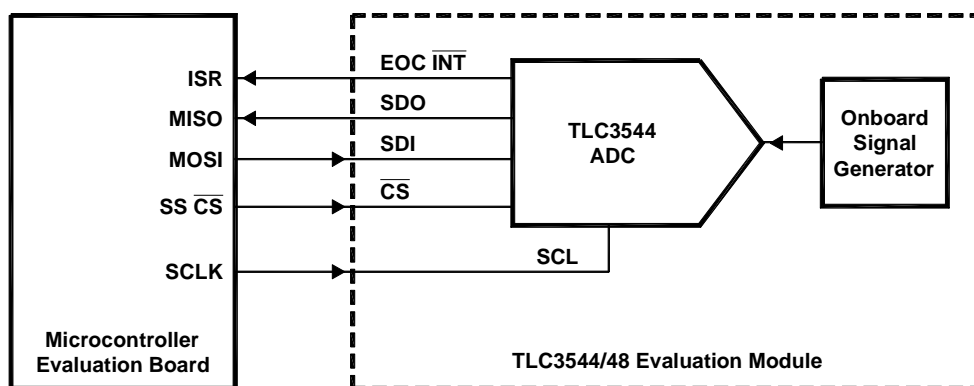


Figure 1. System Configuration

2 TLC3544/48 Evaluation Module

The data converter EVM used for this application note can have either the TLC3544 or the TLC3548 installed at location U9. These are 14-bit, 200-KSPS serial analog-to-digital converters with four (TLC3544) or eight (TLC3548) analog input channels.

The TLC3544 and TLC3548 devices operate from a single 5-V analog supply and a 3.3-V or 5.0-V digital supply. The EVM contains regulators that provide all the necessary voltages from a standard ± 12 -V laboratory power supply.

The EVM features a convenient onboard signal generator as well as all the necessary hardware to provide accurate reference voltages to the data converters. Connectors J12, J13, and J14 are 10-pin dual-row headers. These headers simplify interfacing a microcontroller-based system. The shorting bars can be removed from headers J13 and J14, allowing the user to define a custom signal-interface to the EVM.

System I/O voltages, as well as the voltage on the digital portion of the ADC, can be set to either 3.3 V or 5.0 V by changing the position of W17. When W17 is in position 2-3, the I/O voltage is set to 5.0 V. Position 1-2 on W17, lowers the I/O voltage to 3.3 V, which is necessary when running the low-power MSP430 family of microcontrollers.

Although the onboard signal generator was used as the analog source in this application note, the TLC3544/48 EVM contains a variety of input and output connectors, signal conditioning circuits, and reference voltage options, so that users can define custom analog input conditions. It is even possible to tie the EVM directly into an existing control system – see the TLC3544/48 EVM Users Guide (SLAU073) for complete Details.

Table 1. EVM to Micro Cable Definition

Pin No.	MSP430F149 Port	J12 Pin Name	Signal Description
1	P3.5 - SS \overline{CS}	HOST_CNTLa	Low = ADC \overline{CS} ; High = DAC \overline{CS}
3	P3.3 - CLKS	HOST_CLKXa	Serial clock to EVM
5	Not used	HOST_CLKRa	Serial clock return to HOST
7	P3.1 - MOSI	HOST_DXa	Serial data to EVM
9	P3.2 - MISO	HOST_DRa	Serial data to host
11	P3.6 - GPIO	HOST_FSXa	Frame sync to EVM
13	Not used	HOST_FSRa	Frame sync return to host
15	P1.1 - EINT	HOST_INT*	ADC configured as \overline{INT}
			ADC configured as EOC
17	Not used	HOST_CLKSa	Clock source for host – from EVM
19	P3.7 - GPIO	HOST_CNTLb	\overline{CSTART} to ADC
2-20 (even)	GND	DGND	Ground connections

3 Serial Interface

The following section provides more specific details of the serial data communication pins of the TLC3544 and TLC3548 devices.

3.1 Chip Select (\overline{CS})

Chip select is an active-low input signal. When \overline{CS} is high, the serial-data output (SDO) pin is in a high-impedance state. The serial-data input (SDI) is ignored, and the serial clock (SCLK) is disabled from the internal data clocking circuits. A falling edge of \overline{CS} , resets the internal 4-bit counter, enables SDI, and removes SDO from its high-impedance state. \overline{CS} must be held low for the entire sampling period in order to provide valid data conversions (see literature number SLAS266 for details).

The chip select signal is shown on the interface section of the EVM schematic as HOST_CNTLa. When HOST_CNTLa is pulled low, the TLC3544/48 ADC is selected.

3.2 Serial Data Input (SDI)

The serial data input to the ADC consists of a 16-bit word presented MSB first. The first four MSBs are decoded as a *one-of-four* bit command. The *Configure Write* command (1010b or 0xA) allows the user to enable various operating modes by setting or clearing bits of the internal configuration registers. When initializing the ADC after power up, two write cycles are recommended to ensure correct programming. The first write cycle should contain the CW command, followed by zeros (0xA000h). The second write cycle can contain any valid configuration option as defined in the device datasheet. Input data is shifted into the device on the rising edge of SCLK.

During normal sample/convert cycles, the first four MSBs are responsible for channel selection as well as the selection of various test and FIFO read modes. The remaining 12 bits are normally zeros and are ignored by the ADC.

3.3 Serial Data Output Pin (SDO)

The serial data output from the ADC consists of the 14-bit result of the *previous* conversion cycle. The output data is presented MSB first. When using *chip select* as the conversion trigger, the MSB is available at the falling edge of \overline{CS} . Subsequent data bits are shifted out from the SDO pin on the rising edges of SCLK. Data should be considered valid on the falling edges of SCLK.

3.4 Serial Clock Pin (SCLK)

The host processor provides the serial clock input. SCLK clocks the configuration data into, and conversion data out of, the ADC. When the ADC is appropriately programmed, the serial clock can also be used as the conversion clock. No special configuration is required to run the SCLK in burst mode.

4 Control and I/O Pins

4.1 Conversion Start (\overline{CSTART})

\overline{CSTART} is an active low-external sampling trigger. Pulling this line low initiates the sampling period on the selected (via SDI) analog input. A low-to-high transition starts the conversion process. \overline{CSTART} should be pulled to DVDD when not in use. For the purpose of this application note, a general-purpose I/O pin from the microprocessor was used to hold the \overline{CSTART} pin high.

4.2 Frame Sync (FS)

The FS input signal is normally used in DSP based systems to indicate the start of a serial data frame. If FS is low at the falling edge of \overline{CS} , the rising edge of the frame sync pulse initiates the sample/convert cycle. FS should be pulled to DVDD when not in use.

The TLC3544/48 evaluation module uses a common frame-sync line for both the ADC and DAC. The Xilinx PLD located at U15, acts as a gatekeeper by determining which device (ADC or DAC) should receive the incoming signal. When the HOST_CNTL_a line is low (ADC is selected), the PLD routes the incoming FS to the ADC. When HOST_CNTL_a is high, the FS signal is applied to the DAC.

4.3 End of Conversion/Interrupt (EOC/ $\overline{\text{INT}}$)

The EOC/ $\overline{\text{INT}}$ pin is a programmable output, which indicates the end of conversion (EOC) or acts as an interrupt ($\overline{\text{INT}}$) to host processor. When programmed as EOC, the output goes from a high-to-low state at the end of the sampling period and returns to a high state once the conversion process is complete. In other words, EOC indicates that the ADC is busy doing a conversion.

When programmed as $\overline{\text{INT}}$, the pin goes low after the conversion is complete, and stays low until reset by the falling edge of $\overline{\text{CS}}$ or $\overline{\text{CSTART}}$. A rising edge on the FS input can also clear $\overline{\text{INT}}$.

The assembly code for this application note uses the EOC/ $\overline{\text{INT}}$ pin to signal the processor that the conversion process has completed. The configuration/channel selection information is transmitted to the ADC as two bytes. The processor then enters a loop that transmits zeros or dummy data until it receives an interrupt from the ADC.

4.4 Device Pin-Out

Table 2 shows the complete device pin assignments for both the TLC3544 and TLC3548 devices.

Table 2. I/O Pins of the TLC3544/48

3544	3548	Pin Name	Description
1	1	SCLK	Serial input clock
2	2	FS	Frame sync: DSP frame synchronization input, tied to Vcc when not used
3	3	SDI	Serial data Input: The 4 MSBs selects test modes, mux channel and conversion speed
4	4	EOC $\overline{\text{INT}}$	End of conversion/interrupt: used to indicate the converter is busy, or as ext. interrupt source to host processor
5	5	SDO	3-State serial output of conversion result
6	6	DGND	Digital ground reference
7	7	DVDD	Digital supply voltage: 2.7 to 5.5 V dc
8	8	$\overline{\text{CS}}$	Chip select: active low
9-12		A0 – A4	Analog inputs of the TLC3544
	9-16	A0 – A8	Analog inputs of the TLC3548
13	17	AVDD	Analog supply voltage
14	18	AGND	Analog ground reference
15	19	REFP	Upper reference voltage: 5.0 V maximum
16	20	REFM	Lower reference voltage: nominally ground
17	21	BGAP	Internal band gap compensation: see data sheet for requirements
18	22	AGND	Analog ground reference
19	23	AVDD	Analog supply voltage
20	24	$\overline{\text{CSTART}}$	External start of conversion trigger: used for extended sampling mode

5 ADC Initialization and Operation

5.1 Initializing the ADC

Initializing the ADC is a relatively straightforward task. After power up, EOC/INT is high and the data registers are set to zero. The user has the option of setting a default hardware mode, or programmed mode. Each mode requires two 16-bit initialization cycles. Data output from the power up initialization phase should be ignored. For the purpose of this application note, the initialization cycle consists of chip select being held low through two 8-bit SCLK cycles.

Hardware mode is defined by the status of the SDI pin. After power up, two consecutive initialization cycles put the device into hardware mode if the SDI pin is tied to DVDD. This loads the ADC's configuration register with 0x0800h.

To enter the programmed mode, the host processor must write 0xA000h (WRITE CFR + 000h) to the SDI pin during the first initialization cycle, followed by a WRITE CFR + DATA command during the second.

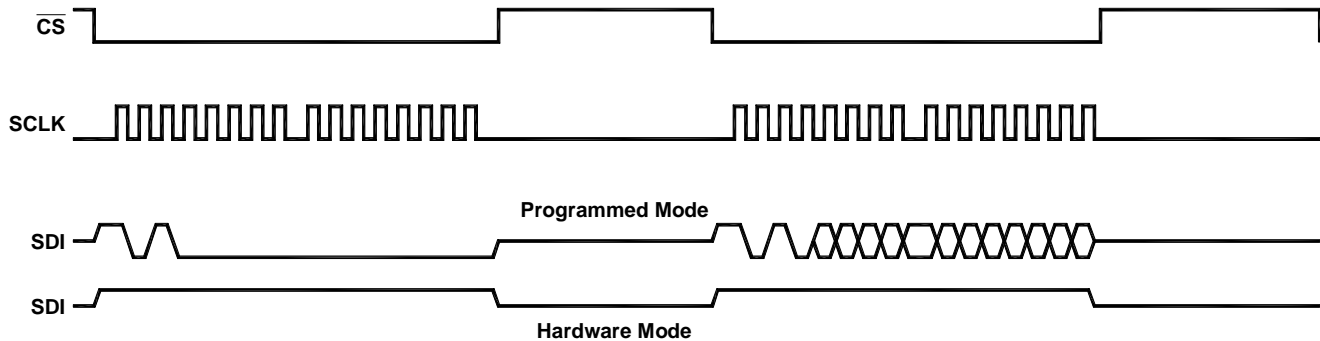


Figure 2. ADC Initialization Cycle

5.2 Operating the ADC

This application note is based on the *one-shot* operating mode (mode 00) using chip select (\overline{CS}) as a conversion trigger. In this mode, each cycle performs one sample and one conversion on the selected analog input channel. Operation begins by bringing chip select low.

Channel selection is accomplished by writing 0000h through 0700h to the SDI pin of the ADC. The microprocessor presented in this application note uses 8-bit data packets when writing to the serial port. The upper and lower bytes are transmitted to the ADC by writing to the micro's data buffer twice, while holding the \overline{CS} pin low. The first byte [D15:D8] contains the channel selection information, and the second byte [D7:D0] consists of zeros, or a dummy value transfer.

The TLC3544 and TLC3548 also feature three test modes that can be accessed by writing 0xB000h, 0xC000h or 0xD000h to the SDI pin. Test mode 1 (0xB000h) presents a digital code on SDO equivalent to $(REFP + REFM) / 2$. Test mode 2 and 3 present a digital code equal to REFM or REFP respectively.

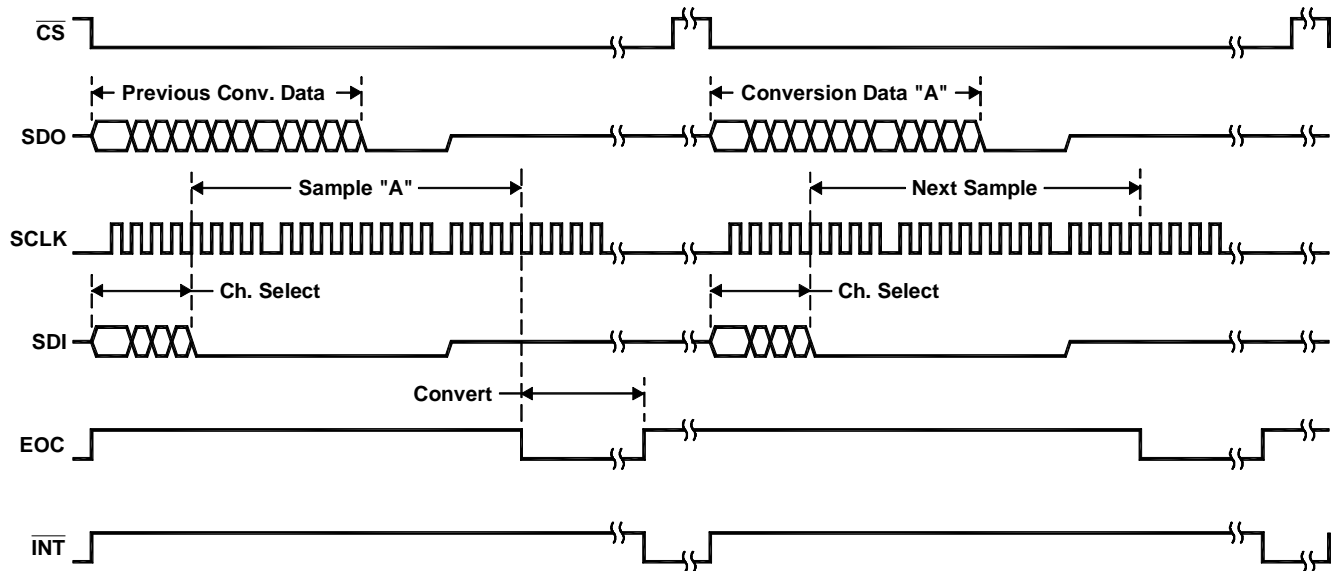


Figure 3. Sample/Convert Cycle

5.3 EOC or \overline{INT}

Figure four shows the relationship of the EOC and \overline{INT} signals more clearly. The EOC signal is active low while the ADC is converting the sampled data. It returns to a high state once the conversion is complete.

\overline{INT} is active low *after* the (sample + conversion) period has finished. \overline{INT} is cleared when a new sample/conversion cycle is initiated by either a falling edge of \overline{CS} (as shown below) or rising edge of FS.

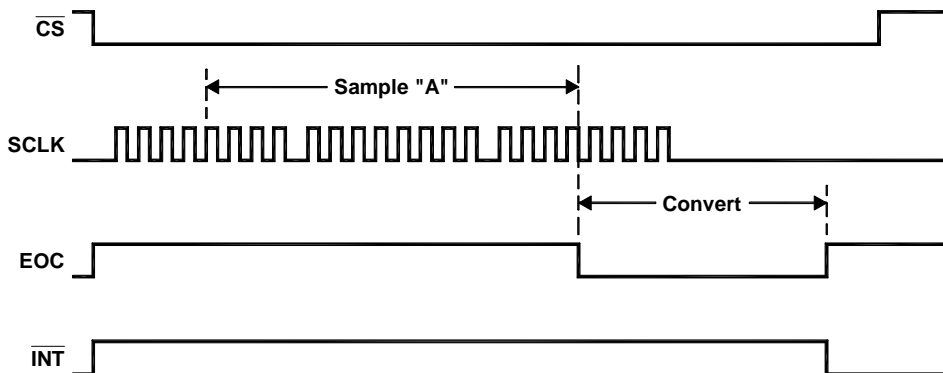


Figure 4. EOC/INT Timing

6 MSPF149 Code Example

The following code example can be downloaded from the web at the following URL:

<http://www.ti.com/sc/docs/psheets/abstract/apps/slaa126.htm>

```

;*****
; MSP430F149 Demo - SPI Communication with TLC3544/48 or TLC3574/78 EVM
; Program implements a digital filter - takes the average of 4 samples
; from CH0 of the ADC and returns them to the TLV5636 DAC
;
; TLV3574/8      MSP430F149
;
; -----
; |          SDI|<---|P3.1      |
; |          SDO|--->|P3.2      |
; CH0~~~>|IN+  SCLK|<---|P3.3    |
; |          /CS|<---|P3.5      |
; |          FS |<---|P3.6      |
; |                                     |
; TLV5636 DAC      |                                     |
; -----
; |          SDI|--->|P3.1      |
; |          FS |<---|P3.6 via PLD |
; |          /CS|<---|P3.5 via PLD |
; |                                     |
;
; Assembled with IAR Embedded Workshop for MSP430 Kickstart
;
; Texas Instruments, Inc.
; Tom Hendrick
; Data Acquisition Applications - Dallas
; Dec. 2000
;*****
#include "msp430x14x.h"    // Standard Equations
#include "TLC357X.h"     // ADC Equations
#include "legal.asm"     // TI Legal Information
;*****
;Constants
;*****
CS      equ  020h    ; Set p3.5 to chip select function
FS      equ  040h    ; Set p3.6 to frame sync function
CSTART  equ  080h    ; Set p3.7 to cstart function
Samples equ  004h    ; Set number of samples

```

```

;*****
;Setup RAM
;*****
        RSEG UDATA0
ADC_Data DS 0          ; Data storage for ADC Samples

;*****
;Setup Stack
;*****
        RSEG CSTACK
        DS 0

;*****
;Program Code
;*****
        RSEG CODE
;*****
RESET_ISR mov    #SFE(CSTACK),SP ; define stackpointer
        call  #Init_Sys      ; Initialize system ports
        call  #SETUP_ADC     ; Initialize ADC
        call  #Mainloop     ; Start Main Loop
Mainloop
        mov.b #Samples, R10  ; Load # of samples in R10
        mov #00, R8         ; Clear R8
SampleLoop
        bic.b #CS,&P3OUT     ; Enable TLC3544/48
        bis.b #01h,&P1OUT    ; Set a test bit - Bit is cleared in ISR

Read_ADC
        mov.b #CH0,&U0TXBUF  ; Dummy write to SPI (generates SCLK)
        call #CLEAR
        mov.b &U0RXBUF,ADC_Data(R8) ; Store Upper Byte
        inc R8              ; Increment data storage pointer
        mov.b #DUMMY,&U0TXBUF ; Dummy write to SPI (generates SCLK)
        call #CLEAR
        mov.b &U0RXBUF,ADC_Data(R8) ; Store Lower Byte
        inc R8              ; Increment data storage pointer

TEST:   mov.b #DUMMY,&U0TXBUF ; Dummy write to SPI
    
```

```

    call  #CLEAR
    bit.b #01h, &P1OUT      ; Continue dummy writes till interrupt
    jnz  TEST
    dec  R10
    cmp  #00,R10
    jnz  SampleLoop        ; Repeat till R10 = 0
;    jmp  Mainloop          ; Repeat
; Remove remark from above line to skip transmit back to DAC

Write_DAC mov #0x0000, R13      ; Clear Register
    mov #Samples, R10
    mov #0x0000, R8           ; Load R8 with 0x0000h
next     mov ADC_Data(R8), R12
    swpb R12                 ; Swap bytes in R12
    and.w #0xFFF0, R12       ; Strip any trailing bits
    rrc.w R12                 ; Shift data 4 places
    rrc.w R12                 ; to conform to
    rrc.w R12                 ; DAC input format
    rrc.w R12
    and.w #0x0FFF, R12       ; Strip any carries
    add.w R12, R13           ; Add shifted data to R13
    incd R8
    decd R10
    cmp #00, R10             ; Is R10 = 0?
    jnz next                 ; Get next sample
    rrc.w R13                 ; Shift result (divide by 4)
    rrc.w R13                 ;
    and.w #0x0FFF, R13       ; Strip any carries
    add.w #0x4000, R13       ; Set DAC Fast Mode
    bic.b #FS,&P3OUT         ; toggle Frame Sync
    bis.b #FS,&P3OUT         ; toggle Frame Sync
    bic.b #FS,&P3OUT         ; to DAC
    swpb R13                 ; Align MSB First
    mov.b R13,&U0TXBUF       ; Transmit upper Data Byte to DAC
    call #CLEAR
    swpb R13                 ; Prepare Lower Byte
    mov.b R13,&U0TXBUF       ; Transmit lower Data Byte to DAC
    call #CLEAR
    bis.b #FS,&P3OUT         ; Set Frame Sync
    jmp  Mainloop           ; Repeat

```

```

;*****
; Clear TX Flag
;*****
CLEAR
    bit.b #UTXIFG0, &IFG1 ; Thank You Eric! TXBUF ready?
    jnc CLEAR             ; 1 = ready
    bic.b #UTXIFG0, &IFG1
    ret
;*****
Init_Sys; Modules and Controls Registers set-up subroutine
;*****
StopWDT  mov    #WDTPW+WDTHOLD,&WDTCTL ; Stop Watchdog Timer

SetupClock
    bic.b #XTOFF, &BCSCTL1
    bis.b #SELM1+SELS, &BCSCTL2
HF_WAIT  ; 8MHz Crystal used - wait for stabilization
    bic.b #OFIFG, &IFG1
    bit.b #OFIFG, &IFG1
    jnz HF_WAIT
    bic.b #OFIFG, &IFG1      ; Clear Oscillator fault flag
    bit.b #OFIFG, &IFG1      ; Test for clear

SetupPort
    bis.b #001h,&P1DIR        ; P1.1 set to Output - Toggles LED
    bis.b #01Eh,&P3SEL        ; P3.1,2,3,4 SPI option select
    bis.b #CS+FS+CSTART,&P3DIR ; /CS, FS & CSTART = output direction
    bis.b #CS+FS+CSTART,&P3OUT ; P3.5,6,7 CS & FS set

SetupInterrupt
    bic.b #02h, &P1IFG        ; Clear Interrupt flags
    bis.b #02h, &P1IES        ; Set for edge selection
    bis.b #02h, &P1IE         ; Enable external interrupt

SetupSPI
    bis.b #040h,&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         ; Set SPI Baud Rate
    mov.b #00h,&U0BR1
    mov.b #00h,&U0MCTL
    eint                      ; Enable interrupts
    
```

```

    ret
;*****
SETUP_ADC ; Initialize the AtoD Converter
;*****

    bic.b #CS,&P3OUT      ; Set ADC /CS Lo
    mov.b #WRITE,&U0TXBUF ; Dummy write to SPI (generates SCLK)
    call #CLEAR          ; Clear SPI Flags
    mov.b #DUMMY,&U0TXBUF ; Dummy write to SPI (generates SCLK)
    call #CLEAR
    bis.b #CS,&P3OUT      ; Set ADC /CS Hi
    bic.b #CS,&P3OUT      ; Set ADC /CS Lo

; Configuration Write to ADC
; See TLC357X.h file for details on the following parameters
    mov.b #(WRITE+SHORT_SAMP+EXT_REF),&U0TXBUF
    call #CLEAR
    mov.b #DUMMY,&U0TXBUF ; Dummy write to SPI (generates SCLK)
    call #CLEAR
    bis.b #CS,&P3OUT      ; Set ADC /CS Hi
    ret
;*****
IRQ_ISR; Exit LPM0 on reti
;*****

    bic.b #01h, &P1OUT
    bis.b #CS, &P3OUT      ; Set ADC /CS High
    bic.b #02h, &P1IFG
    reti                    ; return from interrupt
;*****

COMMON INTVEC      ; MSP430x11x1/MSP430F14x Interrupt vectors
;*****

    ORG    RESET_VECTOR
RESET_VEC DW    RESET_ISR      ; POR, ext. Reset, Watchdog
    ORG    PORT1_VECTOR
PORT1_VEC DW    IRQ_ISR        ; PORT1, Ext. Int.

    END

```

Bibliography

1. *TLC3544, TLC3548 5-V Analog, 3-/5-V Digital 14-Bit, 200-KSPS, 4-/8- Channels Serial ADC With 0–5 V (Pseudo-differential) Inputs*, Data Sheet, Texas Instruments Literature Number SLAS266
2. *TLC3544, TLC3548 Evaluation Module*, Users Guide, Texas Instruments Literature Number SLAU073
3. *MSP430x1xx Family User's Guide*, Texas Instruments Literature Number SLAU049
4. *MSP430x13x, MSP430x14x Mixed Signal Microcontroller*, Data Sheet, Texas Instruments Literature Number SLAS272

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