Application Report
SLAA112 – November 2000

Interfacing the MSP430 and TLC549/1549 A/D Converters

Mark Buccini *Mixed Signal Controllers*

ABSTRACT

This application report describes how to interface an MSP430 mixed-signal microcontroller with TLC549 and TLV1549 3-volt A/D converters. This report is written for the MSP430x11x(1) family, but can be adapted to any MSP430 derivative.

Introduction

Many members of the MSP430 family have integrated A/D converters (ADCs), but in some applications the required analog conversion function is remote, optional, or perhaps an afterthought. In these types of applications, using external ADCs such as the low-cost, easy to use TLC549 and TLV1549 are options. This report demonstrates how to interface an MSP430F1121 to a TLC549. The TLC549 interfaces serially with the MSP430F1121 using three I/O pins with no external components. This application report shows how to interface an MSP430F1121 and TLC549 using bit-by-bit software.

- Only 31 bytes of MSP430 code for the TLC549 driver
- Simple serial interface
- A/D conversion complete in less than 17µs

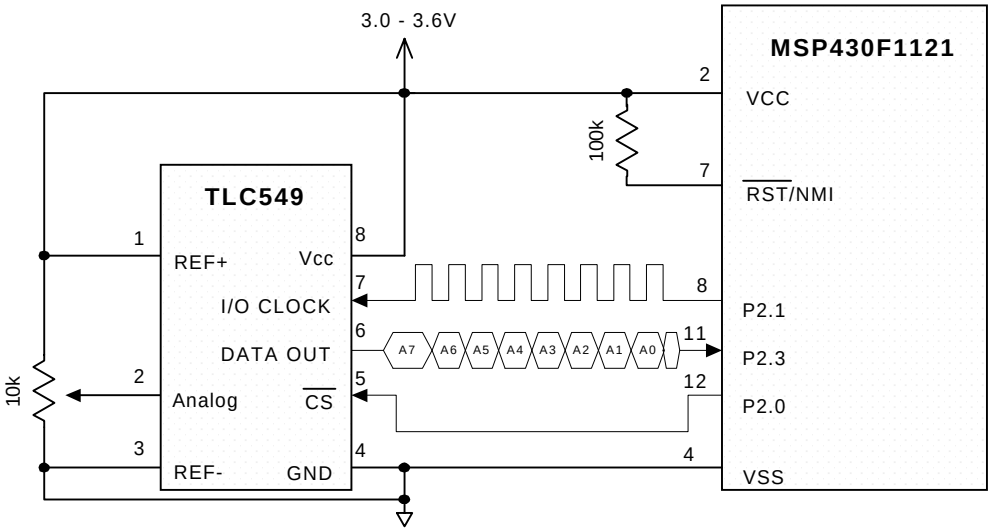

Figure 1. TLC549 – MSP430 Demonstration Circuit

Principle of Operation

Using three digital I/O pins, the MSP430F1121 drives the TLC549 A/D conversion using a synchronous serial interface. In this application report, MSP430 I/O pins P2.0 and P2.1 are configured as outputs using the P2 direction register (P2DIR) and set/reset using the P2 output register (P2OUT). Pin P2.0 interfaces with the TLC549 chip select ($\overline{CS}$) pin P2.1 with the TLC549 input-output clock (I/O CLK). Conversion data from the TLC549 data out (DO) are read on pin P2.3. The choice between P2.0, P2.1, and P2.3 is arbitrary. Any MSP430 I/O pin can be used to drive the TLC549.

When $\overline{CS}$ is high, DO is in a high-impedance state and I/O CLK inactive. To begin the conversion, the MSP430 brings $\overline{CS}$ low. To drive a complete conversion, the MSP430 generates a total of eight clock pulses on P2.1 which are applied to the TLC549 I/O CLK. After $\overline{CS}$ has been brought low, the most significant bit (MSB) from the previous conversion appears on DO. The MSP430 reads the conversion data on DO on pin P2.3 and serially shifts the data into a register ADCDATA (R11). The falling edge of the first four clock pulses shift out the second, third, and fourth most significant bits of the previous conversion. The falling edge of the fourth clock begins the sample function of the analog signal present at the analog terminal of the TLC549. Three more clock pulses are applied to I/O CLK shifting out the least three most-significant bits from the previous conversion. The falling edge of the final (eighth) clock pulse terminates the TLC549 sample function and the hold and conversion cycle begins. The conversion cycle is timed internally by the TLC549 internal oscillator independent of any external clocking, and the conversion is complete in 17uS. $\overline{CS}$ is brought high during the conversion process, and DO returns to a high impedance state. At least 17µs must be allowed before the next conversion sequence or the TLC549 conversion code will be corrupted.

A complete MSP430F1121–TLC549 software example, fet_549.s43, is provided. As coded, the subroutine MEAS_549 requires 150 MCLK cycles and 31 bytes of assembler code including the subroutine call. The entire example requires 60 bytes of code. In the included example, the watchdog is disabled and the stack pointer and MSP430 I/Os after system reset are initialized. The subroutine MEAS_549 is called to drive a TLC549 conversion sequence with the 8-bit conversion code returned in ADCDATA. A register Counter (R12) is also temporarily used to count data bits during the subroutine call. The register definitions for ADCDATA and Counter are arbitrary; any CPU registers, RAM bytes, or even the stack can be used for these registers.

Important: Please review the current MSP430x11x1 and TLC549 datasheets and *MSP430 Users Guide* for device electrical and timing specifications.

Modification for Use With TLV1549

If greater resolution is required, a pin-compatible TLV1549 10-bit converter can be used. One line of the subroutine source MEAS_549 must be modified to support a 10-bit converter. A total of 10 clock pulses are needed and 10 bits of data are shifted into ADCDATA.

```
mov.w   #10,Counter             ; 10 data bits
```

References

1. *MSP430x11x1 Datasheet* (SLAS241)
2. *MSP430x1xx Users Guide* (SLAU049)
3. *TLC548, TLC549 Datasheet* (SLAS067)

Source code for FET_549.s43 Example

```
;*******************************************************************
NAME  fet_549  ; MSP430F1121 - TLC549 Interface Example;
;
#define     ADCData  R11
#define     Counter  R12
P2IN_       equ      00028h                 ; Port 2 Input
P2OUT_      equ      00029h                 ; Port 2 Output
P2DIR_      equ      0002Ah                 ; Port 2 Direction
WDTCTL_     equ      00120h                 ; Watchdog Timer Control
WDTHOLD     equ      00080h                 ; Watchdog hold bit
WDTPW       equ      05A00h                 ; Watchdog password
CS          equ      001h                   ; P2.0 - Chip Select
CLK         equ      002h                   ; P2.1 - Clock
DO          equ      008h                   ; P2.3 - Data Out
;
; Texas Instruments Inc., October 2000
;*******************************************************************
;-------------------------------------------------------------------
            ORG     0F000h                  ; Program Start
;-------------------------------------------------------------------
RESET       mov.w   #0300h,SP               ; Initialize 'x112x stack
StopWDT     mov.w   #WDTPW+WDTHOLD,&WDTCTL  ; Stop Watchdog Timer
SetupP2     mov.b   #CS,&P2OUT              ; /CS set, - P2.x reset
            bis.b   #CS+CLK,&P2DIR          ; /CS and CLK outputs
                                            ;
Mainloop    call    #Meas_549               ; Call subroutine
            jmp     Mainloop                ; Repeat
                                            ;
;-------------------------------------------------------------------
Meas_549;   Subroutine to read TLC549, data is shifted into ADCData
;           (R11), Counter (R12) is used as a bit counter.
;-------------------------------------------------------------------
            mov.w   #8,Counter              ; 8 data bits
            clr.w   ADCData                 ; Clear data buffer
```

```
            bic.b   #CS,&P2OUT              ; /CS reset, enable ADC
ADC_Loop    bit.b   #DO,&P2IN               ; (4) DO -> C (carry)
            bis.b   #CLK,&P2OUT             ; (4) Clock high
            bic.b   #CLK,&P2OUT             ; (4) Clock low
            rlc.w   ADCData                 ; (1) C -> ADCData
            dec.w   Counter                 ; (1) All bits shifted in?
            jnz     ADC_Loop                ; (2) If not --> ADC_Loop
            bis.b   #CS,&P2OUT              ; /CS set, disable ADC
            ret                             ; Return from subroutine
;-------------------------------------------------------------------------
            ORG     0FFFEh                  ;
;-------------------------------------------------------------------------
            DW      RESET                   ; MSP430 RESET Vector
            END
```

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.