Interfacing the TLV1549 10-Bit Serial-Out ADC to Popular 3.3-V Microcontrollers

SLAA005
August 1994

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software 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.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

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

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI 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.

Contents

List of Illustrations

INTRODUCTION

The TLV1549 is a 10-bit serial out analog-to-digital converter that operates from a 3.3-V (±0.3 V) single supply. It uses a switched-capacitor successive-approximation method to perform the conversion in a maximum of 21 μs.

This application report describes how to interface the TLV1549 to three popular microcontrollers which operate from a single 3.3-V supply rail. These are the 68HC05, the TMS70C02, and the 80C51-L.

Interface Timing

The timing for each of the interfaces described in this application report is illustrated in Figure 1. One chip-select ($\overline{CS}$) pulse is used for each 10-bit conversion and 16 CLOCK I/O pulses are between each $\overline{CS}$.

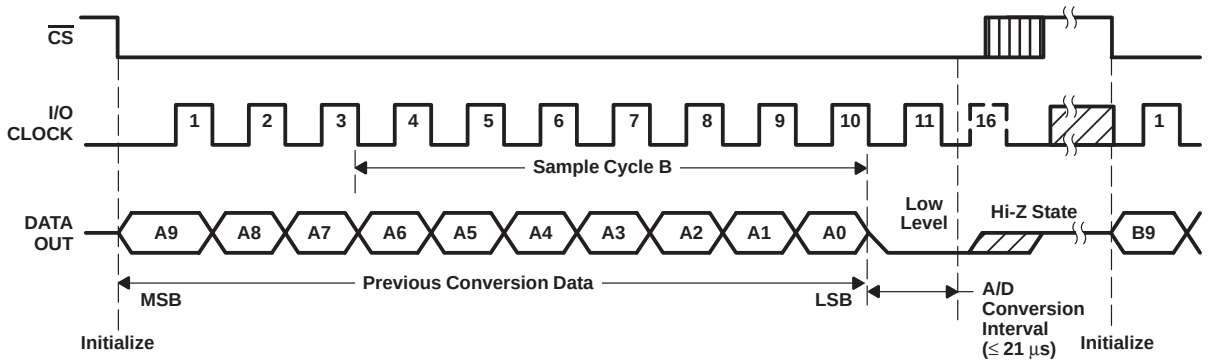


Figure 1. Timing for the 11- to 16-Clock Transfer Using $\overline{CS}$ (Serial Transfer Completed After 21 μs)

Use of Software Subroutines

The subroutines included in this application report have been developed as much as possible as relocatable pieces of software. They include a provision for setting the number of consecutive conversions to be performed. This is either set in the main program as in the TMS7000 and 80C51-L examples or inside the subroutine as used in the 68HC05 example, by programming the number of conversions into COUNT. It is recommended that at least two conversions are performed either after a power-up reset or after a protracted interval between the last conversion. This enables the TLV1549 on-board sample-and-hold to acquire the most recent signal level during the first conversion cycle before converting it into digital form during the second cycle.

Data Format

In whatever format the data arrives at the microcontroller, it is important to ensure that any reformatting, if required, puts the data into a convenient final format. This application report places the most significant byte of the conversion result in one byte of random access memory (RAM) and the least significant byte in an adjacent byte of RAM. The two least significant bits of the 10-bit result are placed in the least significant bit locations of their RAM location.

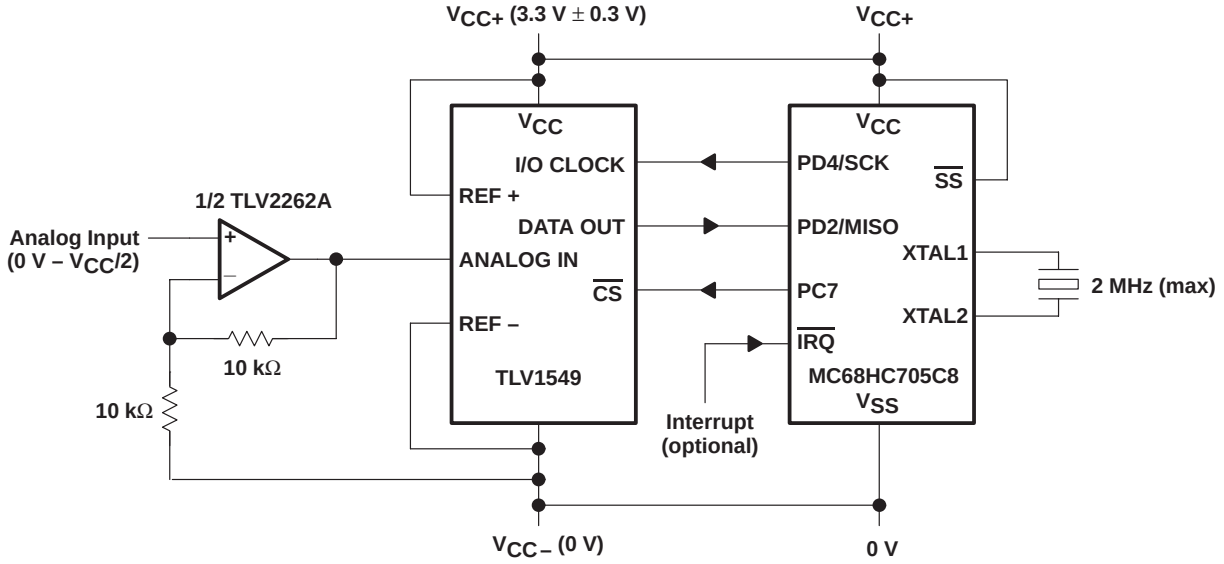
This format gives the user the flexibility to use only 8-bit precision data, if so required, to add the MS and LS bytes together for use in 16 bit wide architectures, to view the two least significant bits of the result for fine tuning applications, or to reformat into another more convenient format.

TLV1549-TO-68HC05 INTERFACE

Microcontroller Features

The M68HC05 family of microcontrollers consists of several different product variants of the basic architecture. It is important that the correct product type is specified to ensure that it contains all the features and attributes necessary to fulfill all its eventual system requirements.

In the case of its suitability for interfacing to the TLV1549 serial out ADC, the M68HC05 product type should contain a serial peripheral interface (SPI). Several types contain this feature including the MC68HC705C8, which was chosen as the target for this ADC interface.



NOTE: For 68HC05 operating off 3.3 V dc supply:
Maximum I/O clock frequency = maximum crystal clock frequency/4 = 0.5 MHz

Figure 2. TLV1549 10-Bit Serial Out ADC-to-MC68HC705C8 Microcontroller Interface

Interface Circuit

Figure 2 shows the circuit interconnections for the TLV1549 – MC68HC705C8 microcontroller interface. No glue logic is required. The positive reference to the TLV1549 is provided directly from the V_{CC+} supply. The analog signal is scaled by an appropriate factor (a gain of two in this case) and buffered by one half of a TLV2262A dual operational amplifier.

The three digital interface terminals, I/O CLOCK, DATA OUT, and $\overline{CS}$ of the TLV1549 connect directly to the PD4/SCK, PD2/MISO, and PC7 terminals respectively of the microcontroller. When the SPI is enabled, PD4 becomes SCK, which is the serial clock output, and PD2 becomes the master in slave out (MISO) terminal. When programmed to be a master device, the microcontroller receives serial data at its MISO terminal.

Figure 3 shows the shift register operation of the SPI when connected to a serial output peripheral component such as the TLV1549. The MC68HC705C8 operates as the master device and the TLV1549 acts as the slave.

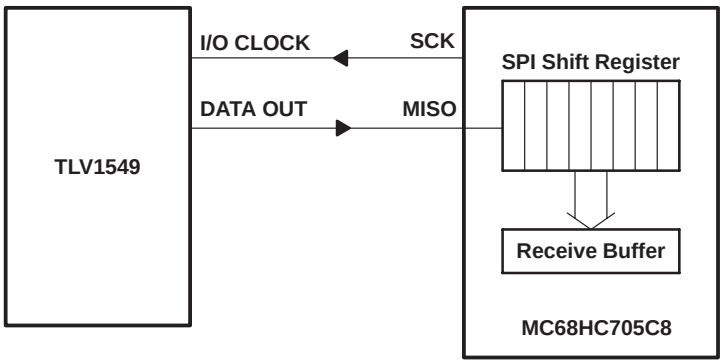


Figure 3. Shift Register Operation of the Serial Peripheral Interface (SPI)

Software Considerations

The three registers which are used for SPI communications are:

Serial peripheral control register (SPCR)
Serial peripheral status register (SPSR)
Serial peripheral data I/O register (SPDR)

Serial Peripheral Control Register (SPCR)

Bits 0 and 1 of the SPCR program the SPI master bit rate. With bits 0 and 1 set to 0, SCK runs at the internal processor clock/2. This means that SCK operates at one quarter of the microcontroller XTAL frequency.

Bit 2 determines the phase relationship between the clock transmitted at SCK and the data appearing on the MISO terminal. If a 0 is placed in bit 3, SCK idles low. This is the correct condition for the TLV1549. A 1 in bit 6 of the SPCR enables the SPI. A 0 in bit 6 disables the SPI. A 1 in bit 4 (MSTR) confers the status of master to the microcontroller.

Serial Peripheral Status Register (SPSR)

The most important bit of the SPSR is bit 7 (SPIF). When set to 1, it indicates that a data transfer between the TLV1549 and the microcontroller has been completed. The SPIF bit is automatically cleared when the SPSR is read and the SPI data register is accessed.

Serial Peripheral Data I/O Register (SPDR)

When the SPIF bit of the SPSR is 1, the SPDR contains the received byte of information from the converter. The contents of SPDR can then be read into a suitable register or location.

Program Listing

The program listing for the TLV1549-to-MC68HC705 interface shown in Figure 2 is included in the following section. COUNT has been set to 2; this ensures that two conversions are performed each time the ADC subroutine is used. The first conversion flushes out potentially erroneous data from the converter output registers. For test purposes, the main program simply performs continuous repeat jumps to the ADC subroutine.

The SPI expects the most significant bit of each received byte to arrive first which is compatible with the order of the TLV1549 output bit stream. This means that no reformatting of the most significant bit of the 10-bit conversion result is required. However, the least significant byte does need to be shifted right by 6 bits.

Program Listing for the TLV1549-to-MC68HC705 Interface

```
                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                    *                                                       *
                    *       TLV1549 – MC68HC705C8 Interface Program         *
                    *                                                       *
                    *  This program contains a subroutine ADC which reads   *
                    *  the serial data from two conversions of the TLV1549  *
                    *  and places the MSByte in address 50H and the LSByte  *
                    *  in address 51H.                                      *
                    *  The data from the first conversion(potentially       *
                    *  erroneous) is overwritten by the result from the     *
                    *  second conversion.                                   *
                    *                                                       *
                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
000A                SPCR      EQU 0AH         * * * * * * * * * * * * * * * *
000B                SPSR      EQU OBH         *                             *
000C                SPDR      EQU 0CH         *                             *
0002                PORTC     EQU 02H         * Names, Peripheral, and      *
0006                DDRC      EQU 06H         * Control Registers            *
0011                SCDAT     EQU 11H         *                             *
000E                SCCR1     EQU 0EH         *                             *
000F                SCCR2     EQU 0FH         *                             *
0010                SCSR      EQU 10H         * * * * * * * * * * * * * * * *
0050                MSBYTE    EQU 50H         * * * * * * * * * * * * * * * *
0051                LSBYTE    EQU 51H         * Names working RAM addresses *
1FFE                RESETH    EQU 1FFEH       *                             *
1FFF                RESETL    EQU 1FFFH       *                             *
0052                COUNT     EQU 52H         * * * * * * * * * * * * * * * *
0160                          ORG 160H        Start Program at 160H
0160  A601                    LDA #01H
0162  C71FFE                  STA RESETH      Load Reset Vector High Byte
0165  A660                    LDA #60H
0167  C71FFF                  STA RESETL      Load Reset Vector Low Byte
016A  CD016F  START     JSR ADC
016D  20FB              BRA START
016F  1E06    ADC       BSET 7, DDRC
0171  A602              LDA #02H
0173  B752              STA COUNT
0175  A610    CONVERT   LDA #10H
0177  1E02    CSHIGH    BSET 7, PORTC   Set Port C bit 7 (TLV1549 CS) high
0179  4A                DECA
017A  26FB              BNE CSHIGH
017C  1F02              BCLR 7, PORTC   Reset TLV1549 CS (Low)
017E  A650              LDA #50H        Load accumulator with 50H
0180  B70A              STA SPCR        Load SPI control register
0182  A600              LDA #00H        Load dummy data into accumulator
0184  B70C              STA SPDR        Receive SPI data
0186  0F0BFD  HBYTE     BRCLR 7, SPSR, HBYTE
0189  B60C              LDA SPDR
018B  B750              STA MSBYTE      Put MSBYTE in Location 50
018D  A600              LDA #00H        Load dummy data into accumulator
018F  B70C              STA SPDR        Receive SPI data
0191  OF00FD  LBYTE     BRCLR 7, 0B, LBYTE
0194  B60C              LDA SPDR
0196  B751              STA LSBYTE      Put LSBYTE in Location 51
0198  3A52              DEC COUNT
019A  B652              LDA COUNT
019C  26D7              BNE CONVERT     If COUNT = 1, do another conversion
019E  A606              LDA #06H        * * * * * * * * * * * * * * * * * *
01A0  98      FORMAT    CLC             *                                   *
01A1  3651              ROR LSBYTE      *      Reformats LSBYTE             *
01A3  4A                DECA            *                                   *
01A4  26FA              BNE FORMAT      * * * * * * * * * * * * * * * * * *
01A6  81                RTS
01A7                    END
```

TLV1549-TO-TMS7000 INTERFACE

Microcontroller Features

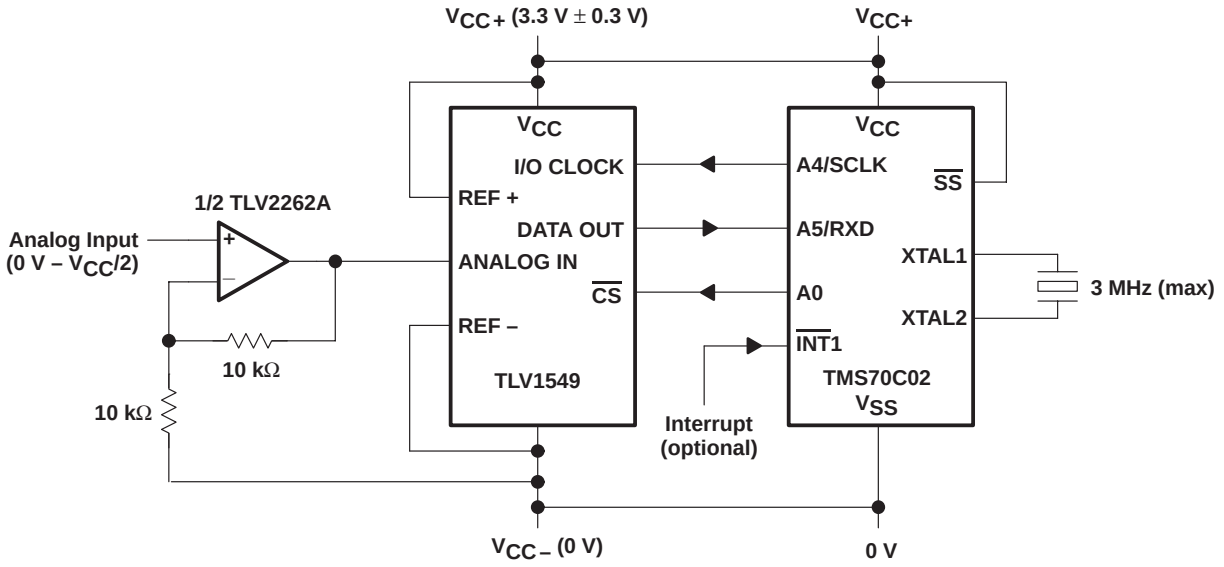
The entire range of TMS7000 microcontrollers can be operated with a 3-V supply. However, the maximum crystal frequency they will tolerate at this supply voltage (over the full temperature range) is 3 MHz. The inherently longer instruction cycle times that this yields should be taken into account when deciding how many software delay loops are necessary to produce the required delay.

Within the family of TMS7000 microcontrollers, three types are available that have a serial port: TMS70Cx2, TMS77C82, and TMS70Cx8. This application report refers to the TMS70Cx2, but any one of these three types could be chosen to efficiently implement a serial interface to the TLV1549.

Three modes of serial communication are available for the serial port: asynchronous mode, isosynchronous mode, and the serial I/O mode. The most suitable of these for interfacing the TMS70Cx2 to the TLV1549 is the serial I/O mode.

Interface Circuit

The TLV1549-to-TMS70C02 interface circuit is shown in Figure 4. The chip select ($\overline{CS}$) of the TLV1549 is controlled by the output from A0 (bit 0 of peripheral port A).



NOTE: Maximum I/O clock frequency = microcontroller crystal frequency/8

Figure 4. TLV1549 10-Bit Serial Out ADC-to-TMS70C02 Microcontroller Interface

Serial I/O Mode

Four peripheral registers are used to set up and control the serial I/O mode of the microcontroller:

Serial mode register (SMODE)
Serial control register 0 (SCTL0)
Serial control register 1 (SCTL1)
Serial port status register (SSTAT)

The contents of the SMODE register determine the data format and type of communication mode (serial I/O for example). In the serial I/O mode, the frame format of each character is five to eight data bits followed by a stop bit. Setting the number of bits to eight can simplify the software necessary to implement the interface.

SCTL0 enables either transmit or receive communication. SCTL1 determines the source of SCLK and programs the frequency of SCLK.

SSTAT is a read-only register that is used for checking the status of the serial port. Bit 1 (RXRDY) of SSTAT is 0 when the receive buffer (RXBUF) is empty and 1 when RXBUF is full.

Provision of TLV1549 Chip Select ($\overline{CS}$)

On power-up and/or system reset, the TLV1549 chip-select terminal ($\overline{CS}$) should be initialized to a high level. To provide this, one of the bidirectional peripheral port bits can be programmed as an output and set to a 1 for a period of at least 21 μs. This period is provided by a delay loop at the beginning of the ADC subroutine. The number of times the loop is excuted in order to achieve at least 21 μs is dependent on the clock frequency of the microcontroller and the number of instruction cycles contained within the delay loop. The example program listing shown in the section program listing for TLV1549-to-TMS70C02 microcontroller interface executes the loop 16 times, but the loop can be executed less times to optimize the conversion throughput rate.

On completion of this delay loop, the particular peripheral port bit is reset to 0, and the converter is now ready to send out data from the previously performed conversion.

Data Reformatting and Storage

After RXBUF is checked to verify it is full, its contents can be read to a suitable register for subsequent access and processing. In the case of the 10-bit conversion result from the TLV1549, two successive bytes of data are received and each are placed in RXBUF to be read consecutively into two convenient memory locations.

The TLV1549 sends the digital result of each conversion with the most significant bit first and the least significant bit last. This is the reverse of the order that the TMS70C02 expects. A few software instructions are therefore inserted near the end of the conversion subroutine that reformat the data into the correct order for interpretation by the microcontroller.

Other Software Considerations

The subroutine that services the TLV1549 conversion should be located in a convenient area of memory that is compatible with the rest of the system. For example, all serial port versions of the TMS7000 family have 8K bytes of EPROM. This EPROM is located between addresses E000H (hex) and FFFFH. A converter subroutine start address at the midpoint of this EPROM memory space may be convenient in that it leaves the first half of this space for the location of the main program. The example program listing in the section Program Listing for TLV1549-to-TMS70C02 Microcontroller Interface uses a start location of F006 which is convenient for the emulation system it was developed on.

On system reset, the stack pointer is at location 0001H. In programs that include nested subroutines where the number of RAM locations taken up by the stack becomes large, the stack can interfere with other useful or even critical RAM locations. It is therefore prudent to reposition the stack pointer, immediately after reset, at a higher address in RAM such as 0060H. This allows the stack plenty of room to grow and avoids interference with lower address RAM locations.

Software Listing

The following program listing reads in the results of two 10-bit conversions from the TLV1549. The software routine ADC actually reads in the results from N conversions, where N is the contents of the register COUNT. The first conversion in a sequence of conversions may be erroneous because the data received is derived from a previous (probably invalid) sample of the analog signal. It is often useful to flush out this first spurious reading before receiving a second valid conversion result. The setting of the contents of COUNT is performed within the main program and should normally be set to a minimum of two.

Program Listing for TLV1549-to-TMS70C02 Microcontroller Interface

```
0001                * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
0002                *           TLV1549 – TMS70C02 Interface Program            *
0003                *                                                           *
0004                *     This program contains a subroutine ADC which reads in *
0005                *     the serial data from two conversions of the TLV1549. The *
0006                *     data (potentially erroneous) from the first conversion *
0007                *     is overwritten by the data from the second conversion. *
0008                *     The most significant byte is placed in register 16. The *
0009                *     least significant byte is placed in register 17.      *
0010                * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
0011       0004     APORT   EQU P4           * * * * * * * * * * * * * * * * * * * *
0012       0005     ADDR    EQU P5           *                                     *
0013       0014     SMODE   EQU P20          *                                     *
0014       0015     SCTL0   EQU P21          *                                     *
0015       0016     SSTAT   EQU P22          *   Name Peripheral Registers         *
0016       0018     SCTL1   EQU P24          *                                     *
0017       0019     RXBUF   EQU P25          *                                     *
0018       001A     TXBUF   EQU P26          *                                     *
0019       0009     COUNT   EQU R9           *   Name Count and Result Registers   *
0020       0010     MSBYTE  EQU R16          *                                     *
0021       0011     LSBYTE  EQU R17          * * * * * * * * * * * * * * * * * * * *
0022  F006                  AORG >F006       Set start address of program
0023  F006   52    START   MOV %>60,B       Set stack register
      F007   60
0024  F008   0D            LDSP
0025  F009   A2            MOVP%>11,ADDR    Set up Port A Data Direction Register
      F00A   11
      F00B   05
0026  F00C   A2            MOVP %>0C,SMODE  Set up Serial Mode Register
      F00D   0C
      F00E   14
0027  F00F   72            MOV %>02,COUNT   Set COUNT = 2
      F010   02
      F011   09
0028  F012   8E            CALL @ADC        Call Subroutine ADC
      F013 F017
0029  F015   E0            JMP START        On return from Subroutine ADC, jump to START
      F016   EF
0030  F017   22    ADC     MOV %>03,A       Put 03 in register A
      F018   03
0031  F019   A2    CSHIGH  MOVP %>01,APORT  TLV1549 Chip Select goes high
      F01A   01
      F01B   04
0032  F01C   B2            DEC A            Decrement the contents of Register A by 1
0033  F01D   E6            JNZ CSHIGH       and jump to CSHIGH if result is not zero
      F01E   FA
0034  FO1F   A2            MOVP %>00,APORT  TLV1549 Chip Select goes low
      F020   00
      F021   04
0035  F022   A2            MOVP %>16,SCTL0  Set up Serial Control Register 0
      F023   16
      F024   15
0036  F025   A2            MOVP %>C0,SCTL1  Set up Serial Control Register 1
      F026   C0
      F027   18
0037  F028   80    LABEL1  MOVP SSTAT,A     Put contents of Serial Status Register in A
      F029   16
0038  F02A   26            BTJO %>2,A,LABEL2  If bit 1 of A is 1, jump to LABEL2
      F02B   02
      F02C   02
0039  F02D   E0            JMP LABEL1       and if not, jump to LABEL1
      F02E   F9
0040  F02F   80    LABEL2  MOVP RXBUF,A     Put contents of RXBUF (MSByte) in A
      F030   19
0041  F031   D0            MOV A,R10        Put contents of A into Register 10
      F032   0A
0042  F033   A2            MOVP %>16,SCTL0  Set up Serial Control Register 0
      F034   16
      F035   15
```

```
0043  F036   A2          MOVP %>C0,SCTL1  Set up Serial Control Register 1
      F037   C0
      F038   18
0044  F039   80   LABEL3  MOVP SSTAT,A     Put contents of Serial Status Register in A
      F03A   16
0045  F03B   26          BTJO %>2,A,LABEL4  If bit 1 of A is 1, jump to LABEL1
      F03C   02
      F03D   02
0046  F03E   E0          JMP LABEL3       and if not, jump to LABEL3
      F03F   F9
0047  F040   80   LABEL4  MOVP RXBUF,A     Put contents of RXBUF (LSByte) in A
      F041   19
0048  F042   D0          MOV A,R11        Put contents of A in Register 11
      F043   0B
0049  F044   D2          DEC COUNT        (COUNT) – 1
      F045   09
0050  F046   E6          JNZ ADC          If COUNT is not zero do another conversion
      F047   CF
0051  F048   B0          CLRC             clear carry bit
0052  F049   DD          RRC R11          * * * * * * * * * * * * * * * * * * *
      F04A   0B                            *                                   *
0053  F04B   E7          JNC LSBIT0       *  Reformats Least Significant Byte   *
      F04C   03                            *                                   *
0054  F04D   74          OR %>2,R11       * * * * * * * * * * * * * * * * * * *
      F04E   02
      F04F   0B
0055  F050   42   LSBIT0  MOV R11,LSBYTE   Put reformatted LSByte in LSBYTE
      F051   0B
      F052   11
0056  F053   D5          CLR R12          clear register 12
      F054   0C
0057  F055   D5          CLR R14          and register 14
      F056   0E
0058  F057   22          MOV %>8,A        Set contents of A to 8
      F058   08
0059  F059   42          MOV R10,R12      Put contents of register 10 in register 12
      F05A   0A
      F05B   0C
0060  F05C   B0   FORMAT CLRC             * * * * * * * * * * * * * * * * * * * *
0061  F05D   DD          RRC R12          *                                     *
      F05E   0C                            *                                   *
0062  F05F   DF          RLC R14          *  Reformats Most Significant Byte    *
      F060   0E                            *                                   *
0063  F061   B2          DEC A            *                                   *
0064  F062   E6          JNZ FORMAT       * * * * * * * * * * * * * * * * * * * *
      F063   F8
0065  F064   42          MOV R14,MSBYTE   Put reformatted MSByte into MSBYTE
      F065   0E
      F066   10
0066  F067   0A          RETS             Return from subroutine ADC
0067  FFFE               AORG >FFFE       Configure Reset vector
0068  FFFE F006          DATA START       to point to START
0069                     END
```

TLV1549-TO-80C51-L INTERFACE

Microcontroller Features

The 80C51-L is the 3.3-V supply version of the 80C51 family of microcontrollers. Various 3.3-V supply versions of the 80C51 architecture are available from different manufacturers. Individual data sheets should be consulted to establish at which maximum crystal frequency each specific device type can operate.

As indicated for the previously described interfaces, the most suitable method of receiving the serial output from the TLV1549 is to configure the serial port of the microcontroller to perform like an 8-bit shift register. The same is true for the 80C51-L.

Serial I/O Mode 0

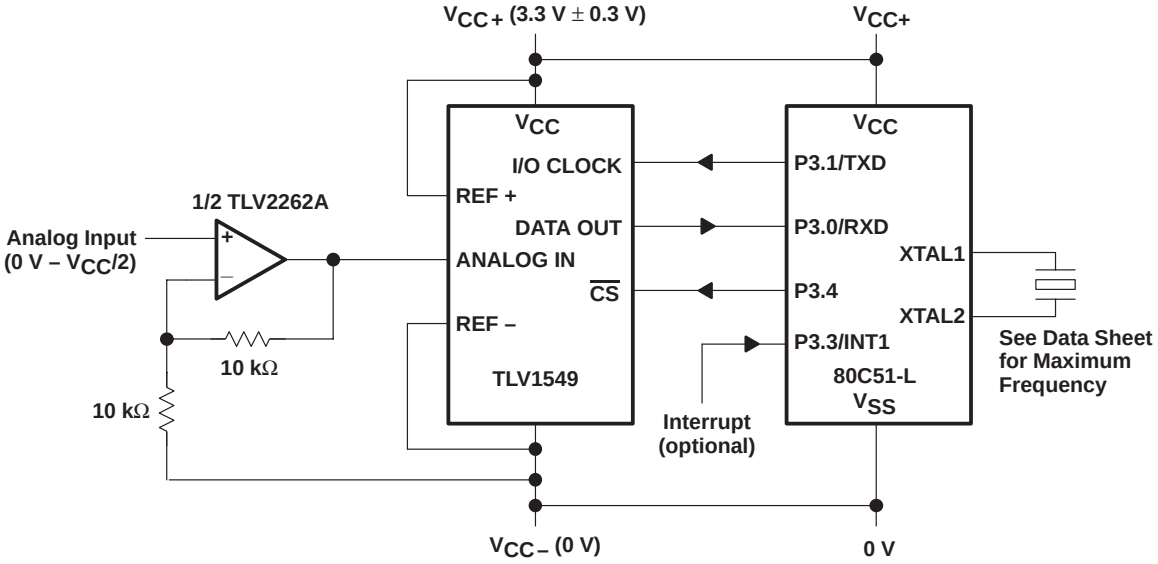
The type of serial communication to and from the 80C51-L is determined by the data inserted into the serial port control register (SCON). The contents of the most significant bits of SCON (bits 7 and 6) determine the operating mode (modes 0, 1, 2, and 3) of the serial port.

Mode 0 is the shift register mode and is programmed by placing a 0 in each of bits 7 and 6 of the SCON. Bit 4 (REN) of the SCON is the receive enable bit. This bit is set to 1, while bit 1 (RI) of the SCON is 0, to receive serial data. In this configuration, data is received at bit 0 of port 3 (P3.0). The synchronizing signal for clocking in this data is output at TXD, which is bit 1 of port 3 (P3.1).

When configured for mode 0, eight bits are received with no trailing stop bit for each enabling of serial reception. The data is received with the least significant bit expected first, the reverse of the order in which the TLV1549 serial data arrives. Reformatting of the received data is therefore necessary.

Interface Circuit

Figure 5 shows the interconnections necessary to implement the interface of the TLV1549 to the 80C51-L microcontroller. $\overline{CS}$ of the TLV1549 is driven by bit 4 of port 3 (P3.4) of the 80C51-L.



NOTE: I/O clock frequency = microcontroller clock frequency/12

Figure 5. TLV1549 10-Bit Serial Out ADC-to-80C51-L Microcontroller Interface

Software Listing

Similar to the previously described program listings, the following listing contains the subroutine ADC that reads into the 80C51-L ten bits of serial data resulting from a single conversion of the TLV1549. The number of consecutive conversions performed for each jump to subroutine ADC is equal to the number placed in COUNT. The result of each conversion is overwritten by that of the next conversion in the sequence.

Program Listing for the TLV1549-to-80C51-L Interface

```
LOC    OBJ    LINE   SOURCE
                 1    ;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                 2    ;*                                                               *
                 3    ;*           TLV1549 – 80C51-L Interface Program                 *
                 4    ;*                                                               *
                 5    ;*     This program contains a subroutine ADC which reads        *
                 6    ;*     in the serial data from the TLV1549 10-bit ADC            *
                 7    ;*     and places the most significant byte in address 20H       *
                 8    ;*     and least significant byte in address 21H                 *
                 9    ;*                                                               *
                10    ;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                11
0020            12    MSBYTE    EQU 20H          ;* * * * * * * * * * * * * * * * * * *
0021            13    LSBYTE    EQU 21H          ;*     Name data destinations        *
REG             14    COUNT     EQU R3           ;*     and COUNT register            *
                15                                ;* * * * * * * * * * * * * * * * * * *
0022            16              ORG 22H           ;Set start address
0022   7B02     17    START:    MOV COUNT, #02H  ;Set COUNT=2 (Do 2 conversions)
0024   020029   18              JMP ADC          ;Jump to subroutine ADC
0027   8OF9     19              JMP START        ;Repeat above again
0029   D2B4     20    ADC:      SETB P3.4        ;* * * * * * * * * * * * * * * * * * *
002B   7410     21              MOV A, #10H      ;*  Set Port3 (bit 4) high          *
002D   14       22    DELAY:    DEC A            ;* (Sets CS of TLV1549 high)        *
002E   70FD     23              JNZ DELAY        :* * * * * * * * * * * * * * * * * * *
0030   C2B4     24              CLR P3.4
0032   759810   25              MOV SCON, #10H
0035   3098FD   26    LABEL1:   JNB SCON.0, LABEL1     ;Read in
0038   C298     27              CLR SCON.0             ;most significant
003A   A899     28              MOV R0,SBUF            ;byte, place in R0
003C   3098FD   29    LABEL2:   JNB SCON.0, LABEL2     ;Read in
003F   C29C     30              CLR SCON.4             ;least significant
0041   C298     31              CLR SCON.0             ;byte,
0043   A999     32              MOV R1, SBUF           ;place in R1
0045   1B      33              DEC COUNT        ;COUNT-1
0046   EB       34             MOV A, COUNT     ;
0047   70E0     35              JNZ ADC          ;If COUNT not = 0,
                36                               ;do another conversion
0049   7C08     37              MOV R4, #08H     ;Put 08H in R4
004B   AA00     38              MOV R2, 00H
004D   C3       39    LOOP:     CLR C            :* * * * * * * * * * * * * * * * * * *
004E   E8       40              MOV A, R0        ;*                                   *
004F   13       41              RRC A            ;*                                   *
0050   F8       42              MOV R0, A        ;*         Reformats MSByte          *
0051   EA       43              MOV A, R2        ;*                                   *
0052   33       44              RLC A            ;*                                   *
0053   FA       45              MOV R2, A        ;*                                   *
0054   1C       46              DEC R4           ;*                                   *
0055   EC       47              MOV A, R4        ;*                                   *
0056   70F5     48              JNZ LOOP         ;* * * * * * * * * * * * * * * * * * *
0058   EA       49              MOV A, R2        ;
0059   F520     50              MOV 20H, A       ;
005B   E9       51              MOV A, R1        ;* * * * * * * * * * * * * * * * * * *
005C   13       52              RRC A            ;*                                   *
005D   F521     53              MOV 21H, A       ;*         Reformats LSByte          *
005F   9209     54              MOV 21H.1, C     ;*                                   *
0061   C20F     55              CLR 21H.7        ;* * * * * * * * * * * * * * * * * * *
0063   22       56              RET              ;Return from subroutine
                57              END
```

ANALOG CONSIDERATIONS

Analog Reference for the TLV1549

The REF+ terminal of the TLV1549 can be directly connected to the V_{CC} rail of the device. This produces accurate results for analog input signals right up to the supply rail. However, if the operational amplifier driving the input is supplied from the same single supply as the ADC, the output of the operational amplifier could possibly be nonlinear up to the rail voltage. If this is a concern, a lower reference voltage as shown in Figure 6 can be applied to REF+ providing more headroom for the amplifier.

The output of the TL2262A 3-V single-supply operational amplifier can swing to within 10 mV of its positive supply rail. This effectively loses only two least significant bits (LSBs) off the top of the digital output range of the TLV1549 when both the amplifier and ADC are powered from the same 3-V supply. The circuit shown in Figure 6 provides a 2.5-V reference to the converter, which restores those bits to the digital output of the TLV1549 while the maximum analog input swing is reduced to 2.5 V.

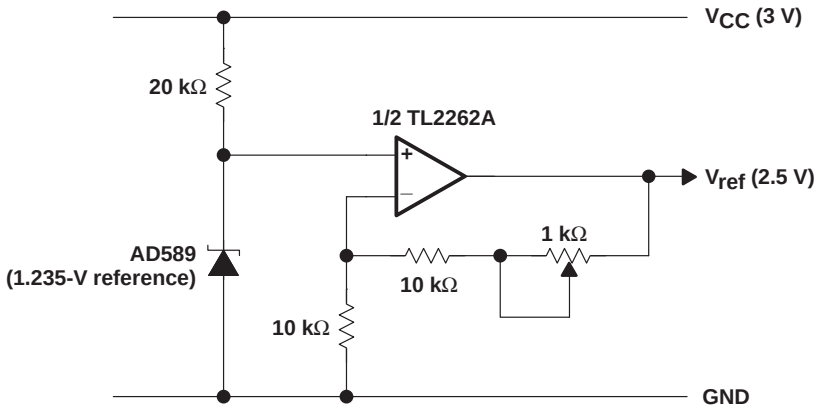


Figure 6. User Adjustable 2.5-V Reference Circuit

PCB Layout

As with all precision analog components, care should be taken in laying out the printed-circuit board (PCB) on which the TLV1549 and chosen microcontroller are placed. The interaction between digital and analog signal paths should be minimized by keeping them as far apart as is physically possible within the constraints of the dimensions of the PCB.

Grounding and Decoupling

Each supply terminal to both the TLV1549 and the microcontroller should be decoupled by a ceramic capacitor of approximately 100 nF in value, situated close to the terminal of the device. Digital and analog ground return paths should be kept separate to prevent any digitally generated currents from corrupting the analog signal.

APPENDIX A

References

MC68HC705C8 Technical Data Manual (1990)	Motorola
M68HC05 Applications Guide	Motorola
TLV1549 Data Sheet	Texas Instruments Incorporated
TMS7000 Family Data Manual (1991)	Texas Instruments Incorporated
Embedded Microcontrollers and Processors Vol. 1	Intel Corporation