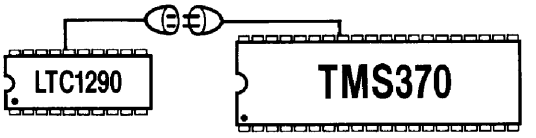
Interfacing the LTC1290/LTC1090 to the TMS370 MCU



Guy Hoover
Sammy Lum
Tim Rust

Introduction

This application note describes an interface between the LTC1290 12-bit data acquisition system and the TMS370 family of microcontrollers (e.g., TMS370C050). The simple four wire interface is capable of completing a 12-bit conversion and shifting data to the TMS370C050 in 42μs. Configuration of the LTC1290 and the TMS370C050 will be discussed as it applies to this interface. Schematics, code, and timing diagrams will be shown. The LTC1090 10-bit data acquisition system is also compatible with this interface. Next the power shutdown feature of the LTC1290 will be discussed and a summary of the key points of this interface will be given, including data throughput rates.

Interface Details

The LTC1290 has two clock lines; ACLK and SCLK. ACLK controls the A/D conversion rate while SCLK controls the data shift rate. Data is transferred in a synchronous, full duplex format over D_{IN} and D_{OUT}.

The TMS370C050 has a Serial Peripheral Interface (SPI) which is a synchronous, full duplex, serial port that allows the user to construct a simple communication path to the LTC1290. SPI provides clock, data in and data out lines that are compatible with the LTC1290. The only additional line required is one programmable output pin (T1EVT) to control $\overline{CS}$ on the LTC1290. The TMS370C050 has two on board timers and one of them can be used to provide ACLK through pin T2IC2/PWM. The schematic of Figure 1 shows how the two devices are connected.

Hardware Description

The code for this interface was developed on a TMS370 application board. The TMS370C050 was used in the microprocessor mode which requires MC (pin 6) be tied high. External memory was used to store the program.

The timing diagram of Figure 2 was obtained with an HP1631A logic analyzer using a 2.5MHz ACLK. The TMS370C050 clock was 20MHz.

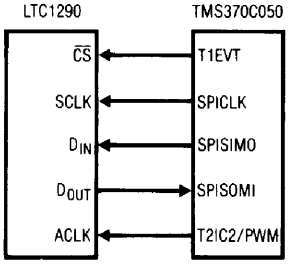


Figure 1. Schematic

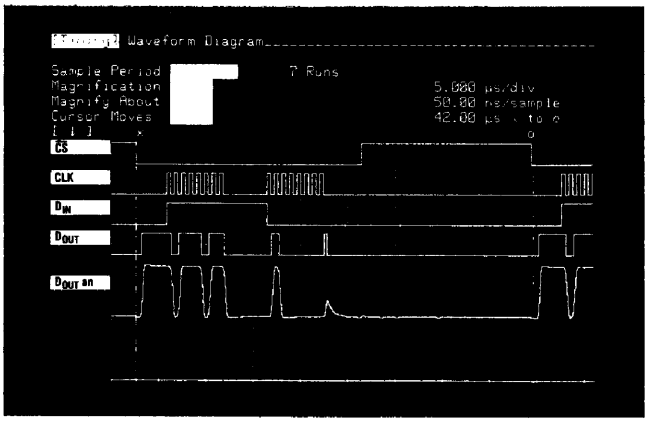


Figure 2. Timing Diagram

The analog section of the schematic in Figure 1 is omitted for clarity. For a complete discussion of the analog considerations involved in using the LTC1290 please see the data sheet.

Software Description

The software configures and controls the SPI of the TMS370C050. Additionally, the software manipulates T1EVT ($\overline{CS}$ of the LTC1290), generates ACLK via pin T2IC2/PWM and generates a delay during which time the LTC1290 performs a conversion.

The System Control and Configuration Control Register 2 (SCCR2) is first configured so the system is operating in the privilege mode. Next the code configures the Timer 2 module to generate a 2.5MHz ACLK. The T2 Compare Registers are loaded with a one. This will generate an ACLK with a frequency equal to one half the internal processor clock (one fourth the crystal frequency). Next the T2IC2/PWM pin is enabled to toggle. Then the pin is configured as an output with the T2IC2/PWM function.

Next the SPI clock is enabled then the SOMI and SIMO functions are enabled. The SPI Configuration Control Register (SPICCR) is set for eight bit character length, an SPI clock that is 1/16 of the crystal frequency, input data transfer on the rising edge, output data transfer on the falling edge and initialization of the SPI. The SPI Operation Contol Register (SPICTL) is set to disable the SPI interrupt, enable transmission and place the SPI in the master mode. Finally the SPI is enabled.

The T1EVT pin on Timer Module 1 is configured as an output pin and $\overline{CS}$ is made to go low or high by placing a "0" or "1" in the T1EVT DATA OUT bit of the Timer 1 Port Control Register 1 (T1PC1). A D_{IN} word that configures the LTC1290 for CH7 with respect to COM, unipolar, MSB first and a 16-bit word length is shown in Figure 3.

T1EVT is made to go low. D_{IN} for the LTC1290 is loaded into the serial data register (SPIDAT). Storing D_{IN} in SPIDAT causes the transfer to begin. After waiting for the first eight bits to be transferred (3 NOP's) the first eight bits containing the MSB's of the LTC1290 are read from the receive data buffer register (SPIBUF) and stored in register 20 as shown in Figure 4. The SPI INT FLAG in the SPICTL register is reset when the data is read from the SPIBUF. The LSB's are transferred in the same manner and are stored in register 21. The data at this point is stored left justified.

1	1	1	1	1	1	1	1
S/D	O/S	S1	S2	UNI	MSBF	WL1	WL0

Figure 3. D_{IN} Word for LTC1290

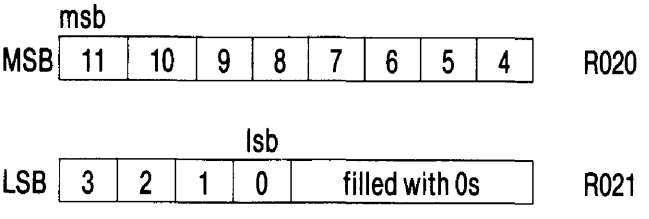


Figure 4. Memory Map

At this time 52 ACLK cycles must be allowed for the A/D to perform its next conversion. Usually the processor will have other tasks to perform during this time. If this is not the case a string of NOP's or a simple delay loop can be used to generate this delay. The code is shown in Figure 5 and a delay loop has been chosen for the A/D conversion.

```
LABEL     MNEMONIC          COMMENTS
          MOV   #001h,P012  ;CONFIGURATION DATA FOR SCCR2

          ;CONFIGURE T2IC2/PWM  PIN FOR ACLK
          MOV   #000h,P062  ;LOAD MSB DATA FOR COMPARE REG
          MOV   #001h,P063  ;LOAD LSB DATA FOR COMPARE REG
          MOV   #070h,P06C  ;CONFIGURATION DATA FOR T2CTL3
          MOV   #030h,P06E  ;CONFIGURATION DATA FOR T2PC2

          MOV   #002h,P03D  ;ENABLE SPI CLOCK
          MOV   #032h,P03E  ;ENABLE SOMI AND SIMO
          MOV   #0CFh,P030  ;CONFIGURATION DATA FOR SPICCR
          MOV   #006h,P031  ;CONFIGURATION DATA FOR SPICTL
          MOV   #00Fh,P030  ;ENABLE SPI
START     MOV   #001h,P04D  ;T1EVT GOES LOW (CS GOES LOW)
          MOV   #0FFh,P039  ;LOAD DIN INTO SPIDAT START SCLK

          NOP               ;3 NOP'S FOR TIMING

          MOV   P037,R020   ;STORE MSB'S IN REGISTER 20
          MOV   #000h,P039  ;START NEXT SPI CYCLE

          NOP               ;3 NOP'S FOR TIMING

          MOV   P037,R021   ;STORE LSB'S IN REGISTER 21
          MOV   #005h,P04D  ;T1EVT GOES HIGH (CS GOES HIGH)

          ;A/D  CONVERSION  DELAY LOOP
          MOV   #002h,R023  ;LOAD DATA IN DELAY COUNTER
DELAY     DEC   R023        ;DECREMENT COUNTER
          CMP   #000h,R023  ;COMPARE COUNTER WITH ZERO
          JNZ   DELAY       ;IF NOT ZERO RETURN TO DELAY
          JMPL  START       ;RETURN FOR NEXT SPI CYCLE
```

Figure 5. TMS370C050 Code

Power Shutdown

The TMS370C050 can be placed in one of three power reduction modes by configuring the SCCR2 register and issuing the IDLE command. Placing the TMS370C050 in the HALT mode reduces the supply current to its minimum power down value of 50μA. Therefore incorporating this with the power shutdown feature of the LTC1290 (typically 5μA in shutdown) it is possible to build a system that draws very low current when not in use. Figure 6 shows such a sequence. An external interrupt is required for the TMS370C050 to exit from the HALT mode. Then three conversion cycles are required to obtain the required data from a measurement and then enter the power shutdown mode again. The data is valid on the second conversion.

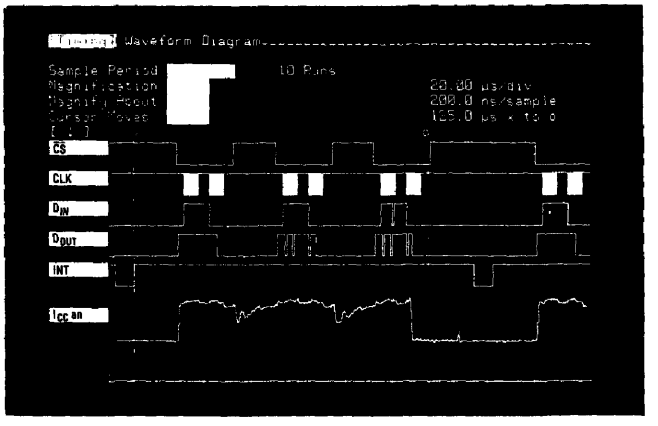


Figure 6. Power Shutdown

The third conversion is required because when power shutdown is requested (D_{IN} = #0FD) the data from the previous conversion is output as a 10-bit word. The bottom trace shows the supply current for the LTC1290 during shutdown and then during conversion. Note there is no wait required for the LTC1290 to start a conversion after exiting from the shutdown mode. The time required for applying an external interrupt, making a measurement, storing the data and entering the shutdown mode is approximately 125μs. The code is shown in Figure 7.

```
LABEL    MNEMONIC          COMMENTS
         .DATA   7FF8h     ;SET INTERRUPT 3 VECTOR WHEN
         .BYTE   70h,00h   ;PROCESSOR COMES OUT OF HALT MODE
         .TEXT   7000h
                           ;INTERRUPT SERVICE ROUTINE EXECUTED
                           ;WHEN PROCESSOR COMES OUT OF HALT
                           ;MODE
         MOV     #005h,P019 ;CLEAR INTERRUPT FLAG
         RTI               ;RETURN FROM INTERRUPT

BEGIN                      ;MAIN PROGRAM
         EINT              ;ENABLE INTERRUPTS
         MOV     #005h,P04D ;CS GOES HIGH
         MOV     #001h,P012 ;CONFIGURATION DATA FOR SCCR2

                           ;CONFIGURE T2IC2/PWM PIN FOR ACLK
         MOV     #000h,P062 ;LOAD MSB DATA FOR COMPARE REG.
         MOV     #001h,P063 ;LOAD LSB DATA FOR COMPARE REG.
         MOV     #070h,P06C ;CONFIGURATION DATA FOR T2CTL3
         MOV     #030h,P06E ;CONFIGURATION DATA FOR T2PC2

         MOV     #002h,P03D ;ENABLE SPI CLOCK
         MOV     #032h,P03E ;ENABLE SOMI AND SIMO
         MOV     #0CFh,P030 ;CONFIGURATION DATA FOR SPICCR
         MOV     #006h,P031 ;CONFIGURATION DATA FOR SPICTL
         MOV     #00Fh,P030 ;ENABLE SPI

START    EINT              ;ENABLE INTERRUPTS
         MOV     #002h,R025 ;SET LOOP REG. TO 2 (WILL LOOP 2 TIMES)

LOOP     MOV     #001h,P04D ;T1EVT GOES LOW (CS GOES LOW)
         MOV     #0FFh,P039 ;LOAD DIN INTO SPIDAT START SCLK

         NOP               ;3 NOP'S FOR TIMING

         MOV     P037,R020 ;STORE MSB'S IN REGISTER 20
         MOV     #000h,P039 ;START NEXT SPI CYCLE
```

Summary

A four wire interface between the LTC1290 and the TMS370C050 with a combined data conversion and data transfer rate of 42μs was demonstrated. The interface used the serial (SPI) port of the TMS370C050. The 12 data bits of the LTC1290 are shifted MSB first in two 8-bit transfers. The data is stored left justified in the TMS370C050's internal registers. By using the power reduction mode of the TMS370C050 and the power shutdown feature of the LTC1290 a low power system was shown to make a measurement in 125μs when active.

```
LABEL    MNEMONIC          COMMENTS
         NOP               ;3 NOP'S FOR TIMING

         MOV     P037,R021 ;STORE LSB'S IN REGISTER 21
         MOV     #005,P04D ;T1EVT GOES HIGH (CS GOES HIGH)

                           ;A/D CONVERSION DELAY LOOP
         MOV     #002h,R023 ;LOAD DATA IN DELAY COUNTER
DELAY    DEC     R023      ;DECREMENT COUNTER
         CMP     #000h,R023 ;COMPARE COUNTER WITH ZERO
         JNZ     DELAY     ;IF NOT ZERO RETURN TO DELAY

         DJNZ    R025,LOOP ;RETURN FOR NEXT CONVERSION

                           ;START POWER SHUTDOWN CONVERSION
         MOV     #001h,P04D ;T1EVT GOES LOW (CS GOES LOW)
         MOV     #0FDh,P039 ;LOAD DIN FOR POWER SHUTDOWN

         NOP               ;3 NOP'S FOR TIMING

         MOV     P037,R022 ;CLEAR SPIBUF
         MOV     #000,P039 ;START NEXT SPI CYCLE

         NOP               ;3 NOP'S FOR TIMING

         MOV     P027,R022 ;CLEAR SPIBUF
         MOV     #005h,P04D ;T1EVT GOES HIGH (CS GOES HIGH)

         MOV     #0C0h,P012 ;PUT TMS370C050 IN HALT MODE
         IDLE

         JMPL    START     ;AFTER EXTERNAL INTERRUPT JUMP TO
                           ;START
```

Figure 7. Power Shutdown Code