

## Interfacing the LTC1290 to the MC68HC05 MCU

Sammy Lum  
 Tim Rust

### Introduction

This application note describes an interface between the LTC1290 12-bit data acquisition system and the Motorola SPI family of single chip microcomputers (e.g., 68HC05). The simple four wire interface is capable of completing a 12-bit conversion and shifting the data to the 68HC05 in 40 $\mu$ s. Configuration of the LTC1290 and the 68HC05 will be discussed as it applies to this interface. Schematics, code, and timing diagrams will be shown. Finally, 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  $\overline{D}_{IN}$  and  $\overline{D}_{OUT}$ .

The Motorola Serial Peripheral Interface (SPI) is a synchronous, full duplex, serial port built into the 68HC05 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 (CO) to control  $\overline{CS}$  on the LTC1290. The schematic of Figure 1 shows how the two devices are connected.

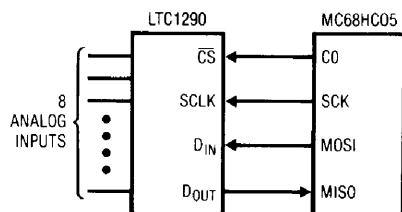
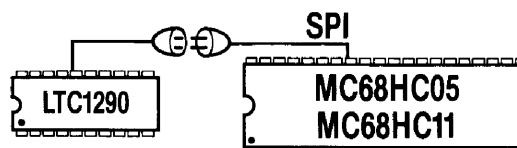


Figure 1. Schematic



### Hardware Description

The 68HC05 was emulated and the code for this interface was developed on a Motorola M68HC05 EVM.

$\overline{SS}$  (Pin 34) of the 68HC05 must be held high to enable the SPI properly for this interface.

The timing diagram of Figure 2 was obtained with an HP1631A logic analyzer using a 4MHz ACLK. The 68HC05 clock was 4MHz.

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.

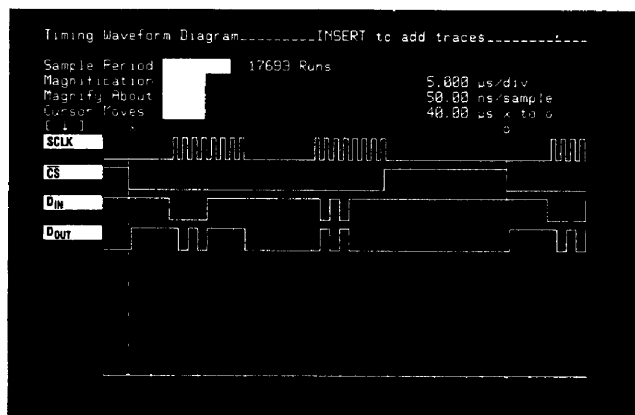


Figure 2. Timing Diagram

### Software Description

The software configures and controls the SPI of the 68HC05. Additionally, the software manipulates CO ( $\overline{CS}$  of the LTC1290) and generates a delay during which time the LTC1290 performs a conversion.

The code first configures the Serial Peripheral Control Register (SPCR) of the SPI. The SPI interrupt is disabled. The SPI outputs are enabled. The SPI is configured as a master. Finally, the SPI clock is set to normally low, for

## Application Note 36B

data transfer on the rising edge and for a frequency equal to half the internal processor clock (one fourth the crystal frequency).

Port C is configured as all outputs by placing ones in the data direction register of port C. A  $D_{IN}$  word that configures the LTC1290 for CH0 with respect to CH1, unipolar, MSB first and a 16-bit word length is stored in memory location \$50. Figure 3 shows how the  $D_{IN}$  word is composed.

CO is made to go low.  $D_{IN}$  for the LTC1290 is loaded into the SPI data register. Storing  $D_{IN}$  in the data register causes the transfer to begin. After waiting for the first eight bits to be transferred (8 NOPs) the status register of the SPI is examined. This clears the SPIF bit of the status register and allows the data register to be read, which is the next step. The first eight bits containing the MSBs from the LTC1290 are then stored in \$61 of the 68HC05 as shown in Figure 4. The LSBs are transferred in the same manner and stored in \$62 of the 68HC05. Notice in Figure 5 that only 6 NOPs are used in transferring the LSBs. This is because after 6 NOPs, time is consumed by the BSET command which sets the CO pin of the 68HC05. The data at this point is left justified.

0	0	0	0	1	1	1	1
S/D	O/S	S1	S2	UNI	MSBF	WL1	WL0

Figure 3.  $D_{IN}$  Word for LTC1290

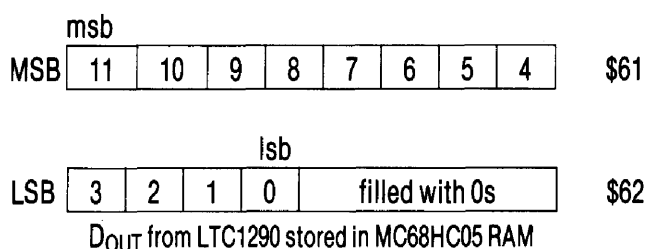


Figure 4. Memory Map

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 NOPs or a simple delay loop can be used to generate this delay.

The code was written for the 68HC05. By changing the addresses of the special function registers however, the code should run on all of Motorola's SPI processors including the 68HC11.

### Power Shutdown

The LTC1290 can be shutdown by inputting the appropriate  $D_{IN}$  word (0D). A dummy conversion prior to a request for power shutdown is required because the data from the previous conversion will be shifted out as a 10-bit word during the power shutdown request. Upon power up the LTC1290 is ready for conversion and the  $D_{OUT}$  word will be valid on the second request for conversion.

### Summary

A four wire interface between the LTC1290 and the 68HC05 with a combined data conversion and transfer time of 40 $\mu$ s was demonstrated. The interface used the serial (SPI) port of the 68HC05. The 12 data bits of the LTC1290 are shifted MSB first in two 8-bit transfers. The data is stored left justified in the 68HC05's internal RAM.

### Reference

Hoover, Guy and Rempfer, William, "Interfacing the LTC1090 to the MC68HC05," Application Note 26B, Linear Technology Corp.

LABEL	MNEMONIC	COMMENTS
	LDA #50	CONFIGURATION DATA FOR SPCR
	STA \$0A	LOAD DATA INTO SPCR (\$0A)
	LDA #FF	CONFIG. DATA FOR PORT C DDR
	STA \$06	LOAD DATA INTO PORT C DDR
	LDA #0F	LOAD LTC1290 $D_{IN}$ DATA INTO ACC
	STA \$50	LOAD LTC1290 $D_{IN}$ DATA INTO \$50
START	BCLR 0,\$02	CO GOES LOW ( $\overline{CS}$ GOES LOW)
	LDA \$50	LOAD $D_{IN}$ INTO ACC FROM \$50
	STA \$0C	LOAD $D_{IN}$ INTO SPI. START SCK
	NOP	8 NOPs FOR TIMING
	LDA \$0B	CHECK SPI STATUS REG
	LDA \$0C	LOAD LTC1290 MSBs INTO ACC
	STA \$61	STORE MSBs IN \$61
	STA \$0C	START NEXT SPI CYCLE
	NOP	6 NOPs FOR TIMING
	BSET 0,\$02	CO GOES HIGH ( $\overline{CS}$ GOES HIGH)
	LDA \$0B	CHECK SPI STATUS REGISTER
	LDA \$0C	LOAD LTC1290 LSBs INTO ACC
	STA \$62	STORE LSBs IN \$62

Figure 5. 68HC05 Code