BURR - BROWN®
APPLICATION BULLETIN
Mailing Address: PO Box 11400, Tucson, AZ 85734 • Street Address: 6730 S. Tucson Blvd., Tucson, AZ 85706
Tel: (520) 746-1111 • Telex: 066-6491 • FAX (520) 889-1510 • Product Info: (800) 548-6132 • Internet: www.burr-brown.com/

INTERFACING THE ADS7870 AND THE MC68HC11E9 ANALOG TO MICROCOMPUTER MADE EASY

By Ed Rojas

INTRODUCTION

Assembly of a data-acquisition system with its many parts (multiplexed, amplifiers, ADC, voltage reference, etc.) can be complex and expensive. The new Burr-Brown ADS7870 solves the problem with its 12-bit low-power CMOS data-acquisition system containing all parts mentioned, and extras. The space saving system is in a small (SSOP-28 surface-mount) plastic package. The price of the entire system is $3.90 (1K Qty).

Here is a simple and versatile two-chip data acquisition solution consisting of a Burr-Brown ADS7870 and a Motorola 68HC11-microcomputer. Serial interfacing is handled through an SPI port. The implementation in this article will demonstrate how to store sixty-four 12-bit conversions in memory of the microcomputer with a sample rate of approximately 20kHz.

ADS7870

The serial interface is designed for ease in interfacing with a wide range of microcontrollers; no glue logic is needed. The serial-interface pins of the ADS7870 are the standard $\overline{CS}$, DOUT, DIN, and SCLK. The interface is compatible with SPI, QSPI, Microwire, and 8051 protocols. There are also various hardware pins that affect the serial interface, or may be used in conjunction with the serial interface, that are listed in Table I.

Analog input to the system is a 4-channel differential, or 8-channel single-ended, multiplexer. Combinations of differential and/or single-ended inputs can be had by a write to the ADS7870 Gain/Mux register. An on-board programmable-gain true instrumentation amplifier with gains of 1, 2, 4, 5, 8, 10, 16, and 20 (see Figure 1). Special features include high input impedance, auto-zeroed for near zero and offset drift, and a PGA status register for overload conditions. A 12-bit SAR Analog-to-Digital Converter (ADC) with a maximum conversion rate of 50kHz, and a precision internal reference (2.048/2.5V) with 10ppm/°C (typical) and good stability. Four serial addressable general-purpose digital I/O pins are provided for any other logic needs.

SERIAL COMMUNICATIONS.		
PIN	**LABEL**	**PURPOSE**
23	$\overline{CS}$	Selects part for Communication and to resynchronize.
22	SDOUT	Serial data output.
21	SDIN	Default (serial data input) with interface control register bit D6 or D1 set LOW (3 wire). If either of the two are HIGH , SDIN is bi-directional (2 wire).
20	SCLK	Serial control input pin.
HARDWARE PINS THAT CAN AFFECT SERIAL COMMUNICATIONS		
9	RESET	Master reset will reset ADC/PGA and SERIAL state machines.
10	R/$\overline{F}$	Sets SCLK polarity.
16	CONV	Start conversion cycle on rising edge of CONV pin. Must remain active HIGH for two CCLK cycles. CONV pin going HIGH will abort any conversion in progress. CONV pin must remain LOW for one clock cycle before going HIGH again.
17	BUSY	Pin HIGH indicates a conversion in progress; can be used to generate an interrupt.
18	OSCE	Enables internal oscillator and drives CCLK pin. also disables D5 (OSCR) and D4 (OSCE) bits of the Reference/Oscillator configuration register.
19	CCLK	Default is an input pin to drive ADC/PGA state machine; can be configured as an output pin driven by the internal oscillator by D4 (OSCE) bit of register 6.
REMAINING PINS.		
1-8	LINEx	Muxed analog input pins. Even (pos) Line 0, 2 ,4 ,6 Odd (neg) Line 1, 3, 5, 7
11-14	DIOx	General-purpose I/O pins. Configured through registers 6, 5.
24	V_{DD}	Supply pin.
25	GND	Ground.
26	V_{REF}	Unbuffered output pin.
27	REF_{IN}	Buffered input pin.
28	REF_{OUT}	Buffered output pin.

TABLE I. ADS7870 Pinout.

Printed in U.S.A. August, 2000

SBAA041

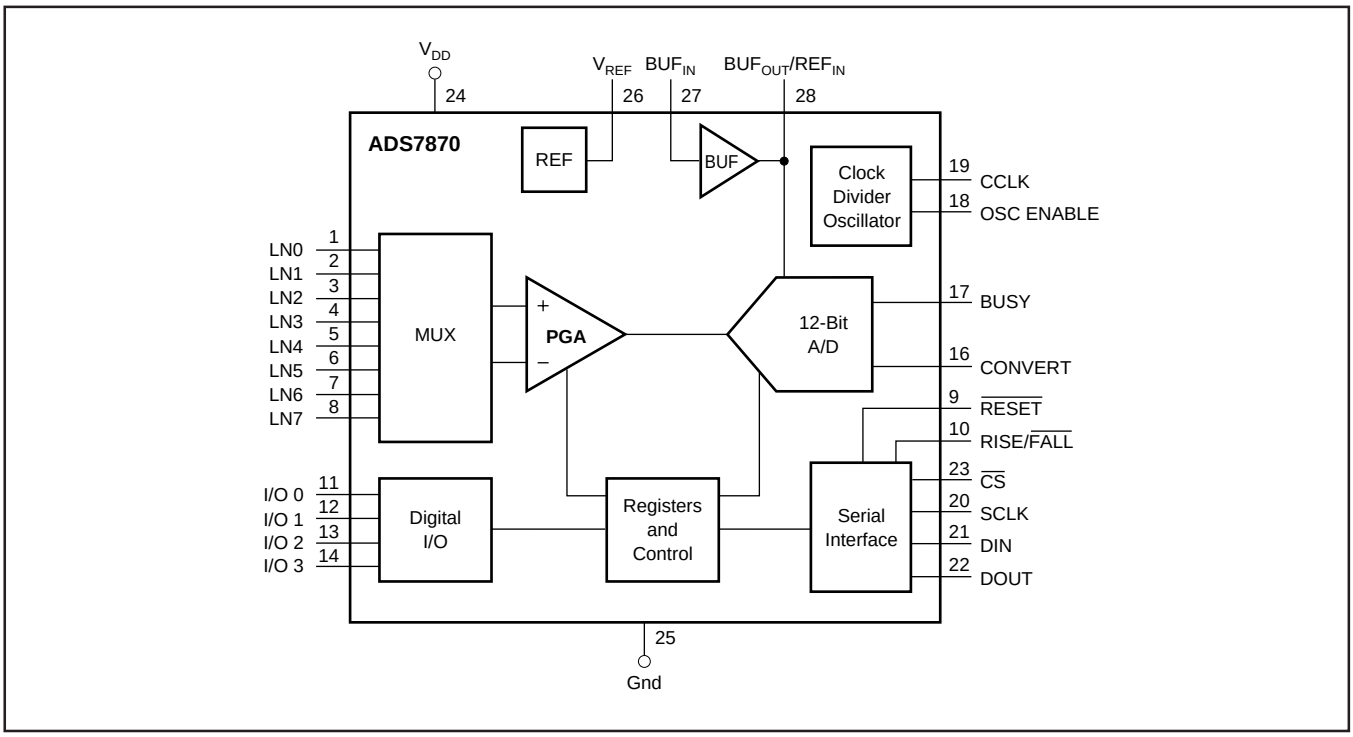

FIGURE 1. ADS7870 System Layout.

MC68HC11 HARDWARE

The interface shown here is for the 68HC11 micro-controller family, specifically the MC68HC11E9. The example test circuit was constructed on Motorola's M68HC11EVBU Universal Evaluation Board. MC68HC11E9 SPI pins are listed in Table II. Configuration of the MC68HC11E9 is single-chip operation with no external memory. The MC68HC11E9 has 512 bytes of EEPROM and 512 bytes of RAM. The program example is stored in EEPROM starting at $B600, which is the jump-to address upon startup if PE0 (pin 43), jp2 of the MC68HC11E9, is tied LOW. The instruction/data array for the ADS7870 is in EEPROM starting at $B7F0, the last 16 bytes of EEPROM memory. 128 bytes of RAM are used to store 64 (12-bit) ADS7870 conversion samples, MSB first, with the 4 lower bits of the LSB byte ignored. The interface program is downloaded to the MC68HC11E9 using its onboard monitor program stored in 12k of ROM. When PE0 (43), jp2, is tied HIGH , a jump to $E000 starts the Monitor program. The Monitor program allows the user to interface with the MC68HC11E9 through a UART (Universal Asynchronous Receiver Transmitter) on the M68HC11EVBU evaluation board. Any PC with Windows® or DOS, and a terminal program can interface through an available COM port.

Windows is a registered trademark of Microsoft Corp.

PIN NO.	LABEL	PURPOSE
22	MISO	Serial Data In
23	MOSI	Serial Data Out
24	SCLK	Serial Output in Master Mode
25	$\overline{SS}$	Output in Master Mode

TABLE II. MC68HC11E9 SPI Pins.

INTERFACE PROGRAM AND DESCRIPTION

Lines 15 – 19 Variables, SPI registers of the HC11.

28 – 37 Data for initialization of the ADS7870 in EEPROM starting at $B7F0. These are the last 16 bytes in user EPROM for up to 8 register writes or reads.

39 – 48 Main program

53 – 64 HC11 set to Master controller. Initializes the SPI port for data direction and data clocking rate.

69 – 88 This initialization subroutine can read from available memory instructions and data bytes to program the ADS7870. By pointing the X index register to the instruction/data array (line 71) and by comparing to the last data address (line 80), the array size can be changed.

92 – 126 Programming the ADS7870 in its auto read-back mode means that after the 8th clock cycle of the instruction 80h (to start a conversion and set PGA gain and mux channels, see Table III) the data from the previous conversion is immediately available for read back. The data, MSB first (or LSB, user choice), is stored in RAM immediately after it is read in and the next conversion started. The SCLK of the HC11 is set to its maximum rate of 1Mhz. The ADS7870 internal clock rate at 2.5MHz. The fact that extra time is needed to store data in RAM gives a maximum sample rate of approximately 20kHz, which is well below the ADS7870 maximum rate of 50kHz.

```
                ********************************************************************
                *  Interfacing program example using M68HC11EVBU & ADS7870EA
                *
                *  This sample program uses the HC11E9 SPI to configure the
                *  ADS7870 and process 64 conversion samples.
                *  Program & data is loaded in HC11E9 EEPROM. Conversion samples
                *  are stored in HC11E9 RAM.
                ********************************************************************
                *  HC11E9 Registers
                ********************************************************************
                PORTD     EQU $1008     * PORT D data register
                DDRD      EQU $1009     * PORT D data direction
                SPCR      EQU $1028     * SPI control register
                SPSR      EQU $1029     * SPI status register
                SPDR      EQU $102A     * SPI data register
                ********************************************************************
                *  Program data codes for ADS7870 stored in EEPROM
                *  codes are loaded manually starting at $B7F0
                *  timing diagram for 8bit write (see Figure 5)
                ********************************************************************
                ORG       #$B7F0        * data array address

                FCB       #$00          * software reset (39 zeros then a one)
                FCB       #$00          * can be used to eliminate CS connection
                FCB       #$00
                FCB       #$01
                FCB       #$03          * 1 byte write to ADS7870 register 3
                FCB       #$04          * data to register 3, set bit RBMO (auto read-back mode)
                FCB       #$07          * 1 byte write to ADS7870 register 7
                FCB       #$1C          * data to register 7, set bit OSCE, REFE, BUFE
                ********************************************************************
                *  Main Program
                ********************************************************************
                ORG       #$B600        * Optional program start point.

                LDS       #$01C8        * start of stack
                LDY       #$1000        * point Y index to SPI register

                BSR       INIT_SPI      * call initalizion subroutine
                BSR       INIT_ADS      * call ADS7870 initialization subroutine
                BSR       SAM_64        * call ADS7870 sampling subroutine

                JMP       $E000         * jump to BUFFALO monitor program

                ********************************************************************
                *  Subroutine: Initialize HC11 SPI port
                ********************************************************************
INIT_SPI        PSHA                    * A index register to stack
                LDAA      #$2F          * x x 1 0, 1 1 1 1
                                        * SS-HIGH, SCLK-LOW, MOSI-HIGH
                STAA      PORTD         * SS stays HIGH , DDRD5 set to 1
                LDAA      #$38          * X X 1 1,  1 0 0 0
                STAA      DDRD          * SS, SCLK, MOSI - Outs
                                        * MISO, TxD, RxD - Ins
                LDAA      #$54          * 0 1 0 1, 0 1 0 0
                STAA      SPCR          * SPI on of HC11 Master, CHPA-1, CPOL-0,
                                        * E/2 SCLK rate, SPR1-0, SPR0-0
                PULA                    * restore register A
                RTS                     * return to main
```

```
65                  ********************************************************************
66                  *  Subroutine: Initialize ADS7870 (refer to ADS program data in HC11
67                  *  program Lines 30-37)
68                  ********************************************************************
69      INIT_ADS    PSHX                        * X index register to stack
70                  PSHA                        * A index register to stack
71                  LDX         #$B7F0          * set index register to first instruction
72                  BCLR        $08,Y $20       * set CS LOW, synchronize ADS serial interface
73
74      INSTRU_L    LDAA        $00,X           * load first instruction register A
75                  STAA        SPDR            * clock data out of HC11 SPI port
76            W1    LDAA        SPSR            * check SPI status register
77                  BPL         W1              * if SPSR bit 7 set transfer complete
78
79                  INX                         * increment to ADS data byte
80                  CPX         #B7F8           * compare index register to address after last
81                                              * byte in instruction and data string
82                  BNE         INSTRU_L        * if programming of ADS complete continue
83
84                  BSET        $08,Y $20       * set CS HIGH, disable ADS serial interface
85
86                  PULA                        * restore register A
87                  PULX                        * restore register X
88                  RTS                         * return to main
89                  ********************************************************************
90                  *  Subroutine: HC11 initiates ADS7870 conversion and reads MSB first.
91                  ********************************************************************
92        SAM_64    PSHX                        * X index register to stack
93                  PSHA                        * A index register to stack
94                  LDX         #$0100          * point index register X to conversion data array
95                                              * in RAM
96                  BSET        $08,Y $20       * set CS HIGH, disable ADS serial interface
97                  BCLR        $08,Y $20       * set CS LOW, synchronize ADS serial interface
98
99        DATA_L    LDAA        #$80            * 1 0 0 0, 0 0 0 0
100                 STAA        SPDR            * D7-HIGH start a conversion
101                                             * Gain 1 D6-D4 LOW, Mux input 0 and 1 D3-D0 LOW
102           W2    LDAA        SPSR            * check status register
103                 BPL         W2              * check for transfer complete
104
105                 STAA        SPDR            * send don't care data byte
106           W4    LDAA        SPSR            * load status register
107                 BPL         W4              * check for transfer complete
108
109                 LDAA        SPDR            * load MS byte, ADS previous conversion
110                 STAA        $00,X           * store in data array
111                 INX                         * increment to next data address
112
113                 STAA        SPDR            * send don't care data byte
114           W5    LDAA        SPSR            * load status
115                 BPL         W5              * check for transfer complete
116
117                 LDAA        SPDR            * load LS byte, ADS previous conversion
118                 STAA        $00,X           * store LS byte
119                 INX                         * increment array, 12-bit conversion acquired
120
121                 CPX         #$0182          * compare to next address after last in array
122                 BNE         DATA_L          * if 64 12-bit transfers complete, continue
123
124                 PULA                        * restore register A
125                 PULX                        * restore register X
126                 RTS                         * return to main
```

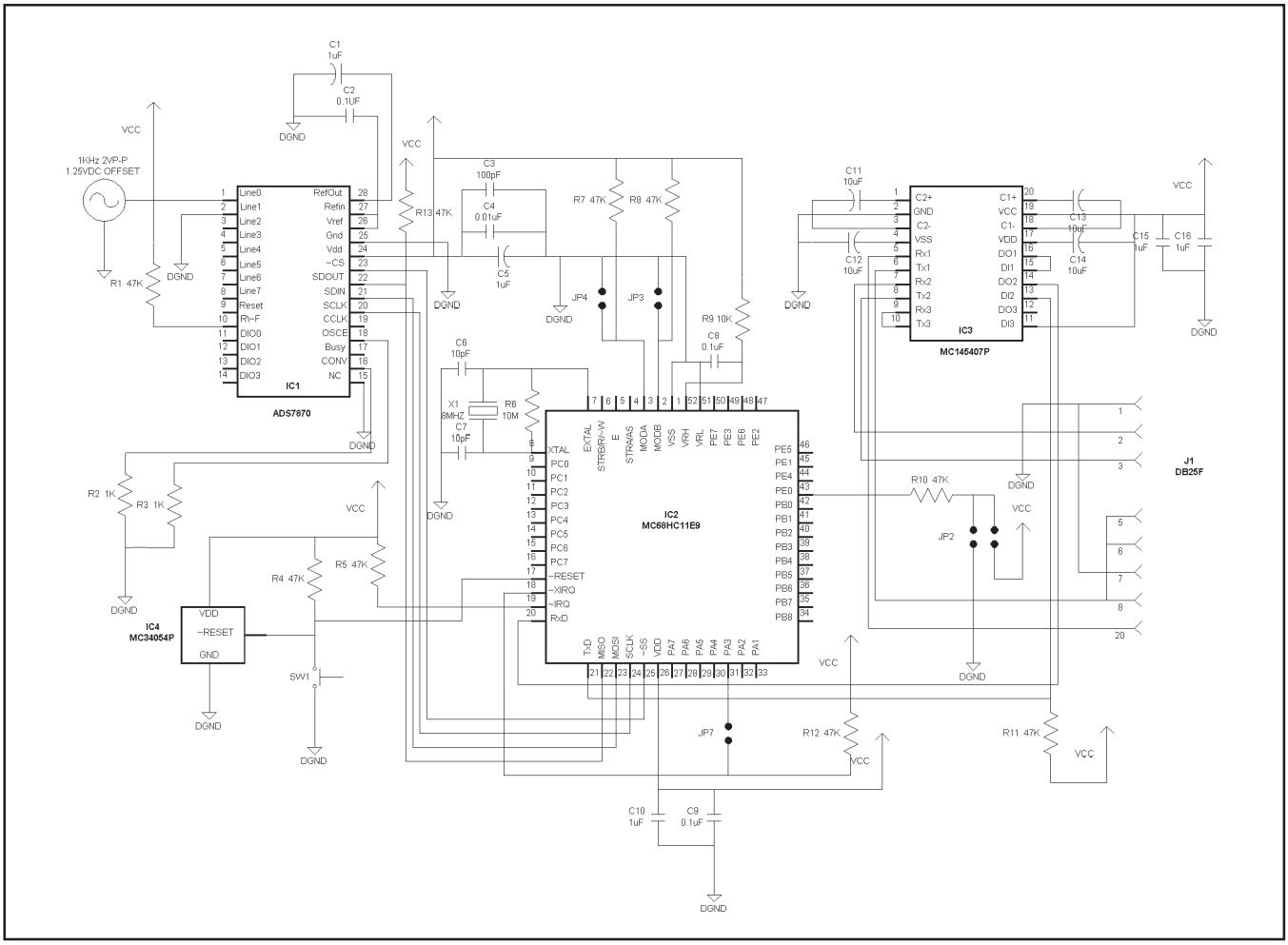

FIGURE 2. ADS7870 and MC68HC11E9 Data Aquisition System.

WIRING THE SYSTEM CIRCUIT

The M68HC11EVBU has all the necessary hardware and software for interfacing with the PC. The ADS7870 comes in an SSOP package. A 28-pin Enplas (OTS-34-0.65-01) socket will ease the breadboarding procedure. After wiring the hardware as shown in Figure 2, adding the jumpers jp4 and jp7 will place the MC68HC11E9 in the single-chip mode of operation. All program code, instructions, and data conversion will reside on the microcontroller. For use with M68HC11EVBU Monitor program, connect jp2 to GND, and leave jp3 open.

RUNNING HARDWARE FROM PC

With the jumpers in place connect the M68HC11EVBU to a PC, start the PC terminal program, and set modem port settings according to the evaluation board manual. Power up the ADS7870 along with EVBU, or after power up push the reset button on the EVBU. A message from the Monitor program indicating program version will confirm that a link has been made. Assembly and debugging can all be done from the Monitor program. Most link-up problems will most likely be in the setup of the PC terminal program and serial communications port; a shared PC port may cause problems. It is best to choose a port that will not be shared by other PC software or hardware configurations.

The example program was used to acquire the data in Figure 3. Using a logic analyzer to view the output from subprogram SAM_64 (line 92), the patterns should match the timing diagram in Figures 4 and 5. Before any data conversions are verified, a simple test will verify the serial communications link between the ADS7870 and the MC68HC11E9. A single data-byte write to digital I/O

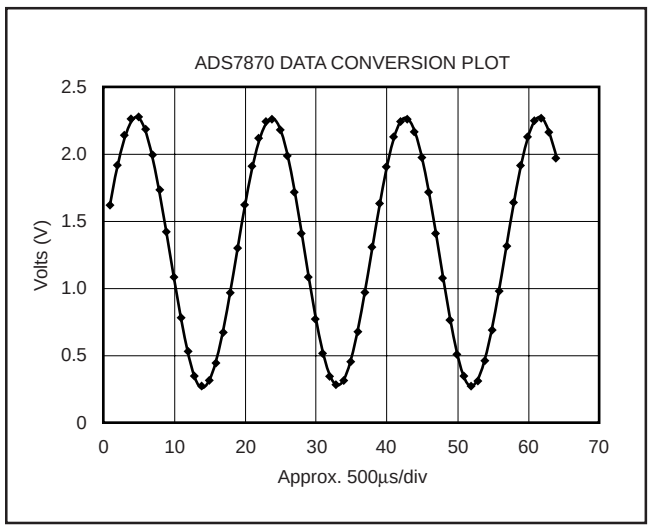

FIGURE 3. ADS7870 Data Conversion Plot.

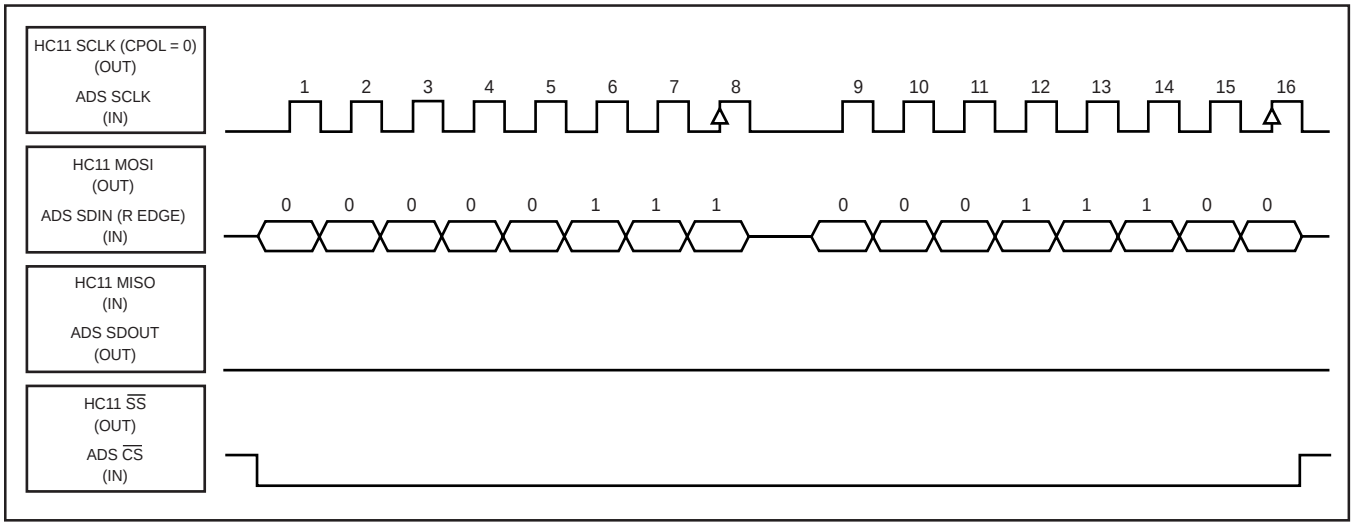

FIGURE 4. 1-Byte Write Instruction (07h), Followed by Data Byte (1Ch).

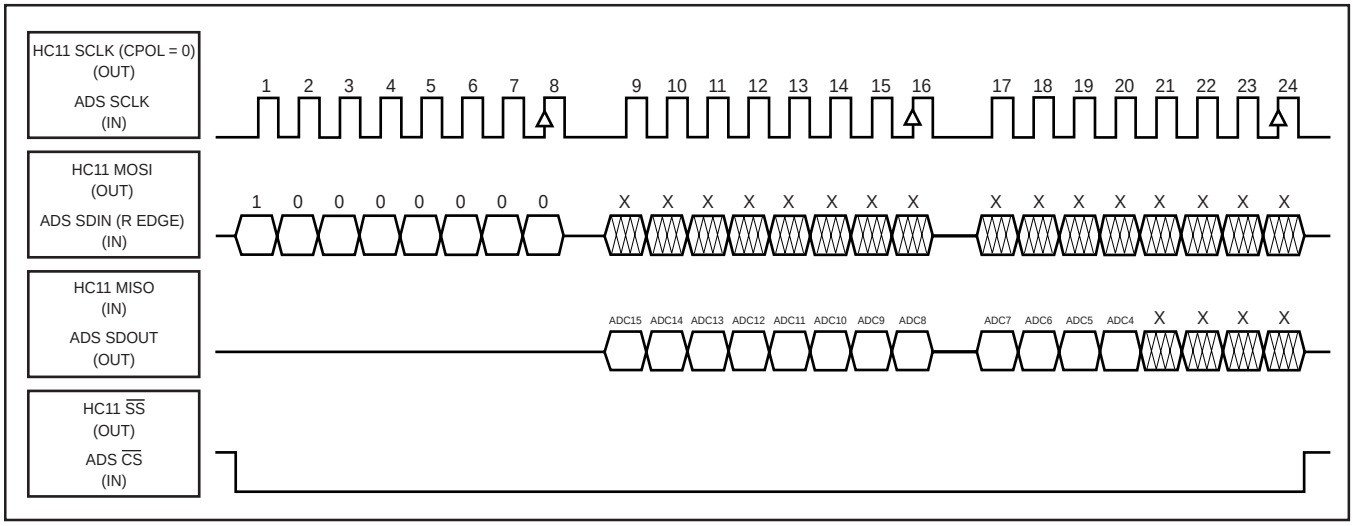

FIGURE 5. 1-Byte Conversion Start and Gain/Mux Register Write (80h), Followed by a 12-Bit Auto Read Back.

control register instruction/address 06h sent first, followed by the data byte 0Fh, will enable the I/O pins as outputs. A second write to the digital I/O register instruction/address 05h, and data byte 0Fh will turn on the digital I/O output drivers.

The instructions and data can be added to the ADS7870 initialization subroutine in line 74 by advancing the address four bytes. The instructions and data must be entered manually by modifying the memory locations in the instruction/data array starting at $B7F0.

The ADS7870 interfacing program (without the program lines numbers) can be downloaded and assembled with the AS11.EXE and PCBUG11.EXE provided with the M68HC11EVBU or found on the Motorola web site. The program can also be entered one line at a time using the Motorola (BUFFALO) Monitor program. Addressing for labels must then be calculated and entered.

The hardware, as shown in the schematic of Figure 2, and the program example will function with $\overline{CS}$ controlled by the micro-controller or by connecting $\overline{CS}$ to digital ground. The program and hardware will continue to operate the same in both configurations with no change in sample rate. When the $\overline{CS}$ of the ADS7870 is connected to the micro-controller, the resynchronization is done once at the start of subroutine SAM_64 (line 96-97); if connected to digital ground, resynchronization is done in subroutine INIT_ADS through the serial interface.

DATA COLLECTION

Run the BUFFALO program and download ADS7870 conversion data to terminal program using the memory dump command to view results. A screen capture (when using the Windows-based terminal program) is an easy, although tedious way of importing the data into a spreadsheet. After some number conversions and scaling, results such as the plot in Figure 3 can be attained.

SUMMARY

The outline in Table III is useful in further evaluation of the ADS7870. Serial instructions and internal registers are listed. Detailed information on the ADS7870 can be found at the Burr-Brown web page (http://www.burr-brown.com). The hardware and software will provide a starting point in the evaluation of the ADS7870, or the basis for a space-saving and low-cost data-acquisition system.

REGISTER ADDRESS					ADDR N0.	READ/ WRITE	REGISTER ADDRESS								REGISTER NAME
AS4	AS3	AS2	AS1	AS0			D7(MSB)	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	Read	ADC3	ADC2	ADC1	ADC0	0	0	0	OVR	A/D Output Data, LS Byte
0	0	0	0	1	1	Read	ADC11	ADC10	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	A/D Output Data, MS Byte
0	0	0	1	0	2	Read	0	0	VLD5	VLD4	VLD3	VLD2	VLD1	VLD0	PGA Valid Register
0	0	0	1	1	3	R/W	0	0	0	0	RBM1	RBM0	CFD1	CFD0	A/D Control Register
0	0	1	0	0	4	R/W	CNV/BSY	G2	G1	G0	M3	M2	M1	M0	Gain/Mux Register
0	0	1	0	1	5	R/W	CNV/BSY	0	0	0	IO3	IO2	IO1	IO0	Digital I/O State Register
0	0	1	1	0	6	R/W	0	0	0	0	OE3	OE2	OE1	OE0	Digital I/O Control Register
0	0	1	1	1	7	R/W	0	0	OSCR	OSCE	REFE	BUFE	R2V	RGB	Ref/Oscillator Control Register
1	1	0	0	0	24	R/W	LSB	2W/3	8051	0	0	8501	2W/3	LSB	Serial Interface Control
1	1	1	1	1	31	Read	0	0	0	0	0	0	0	1	ID Register

TABLE III. Register Address Map.

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.