Popular Electronics
SEPTEMBER 1978

BUILD A LOW-COST A/D CONVERTER

INTERFACING a digital computer with the "real" world requires some means of converting analog (slowly varying) signals into a digital form that can be used by a computer. The low-cost ($30) analog-to-digital (A/D) converter described here can accept up to four channels of analog data, spanning from 0 to +2 volts dc, and change this information into 3½ digits of BCD data.

With such a converter, a computer need not be limited to keyboard entry for many game programs. Now, joysticks or potentiometers can be used. And such real-world sensing of variables like voltage, current, temperature, frequency, and various levels of acidity, salinity, and chemical concentrations can make your computer a powerful and versatile controller. As a bonus, the A/D converter, becomes a powerful test instrument for circuit design and troubleshooting. In this application, up to four channels of voltage, current, and resistance can be monitored with proper input adapters.

Technical Details. The converter produces five conversions per second. It has four input channels and 3½ digits of BCD data output. It is also TTL compatible in input and output and will work with any 8-bit computer that has a latched output port and a three-state input port. Digit and channel selection is under software control. Since the circuit is all CMOS, very little power is required.

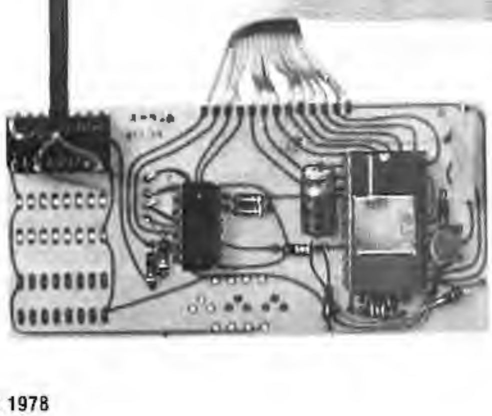
As shown in Fig. 1, the A/D converter employs two IC's and a handful of passive components. One of the four input switch *IC2* to form the input for A/D converter *IC1*. The analog switches are set by data written out by the latched output port of the computer. Resistors *R6* through *R9* provide pullup for the analog switch select lines.

A/D converter *IC1* is a pulse-modulation type. Its chip contains the conversion circuitry, an addressable digit latch, multiplexer, BCD encoder, and system clock.

Conversion control, output digit select, and the output latch are connected to the computer's output port. Data written to this port controls the data placed on the four output lines of *IC1*. The four data output lines from *IC1* are connected to the computer's three-state input port's lower four bits (Dϕ through D3). The upper four bits (D4 through D7) are grounded.

Trimmer potentiometer *R1* deter-

(Continued on page 47)



Two-chip, four-channel converter works with any 8-bit computer.

BY W. L. GREEN

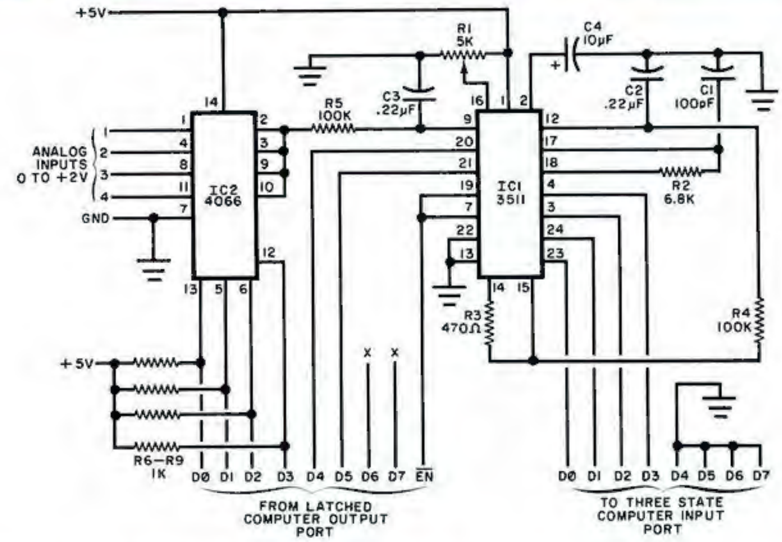
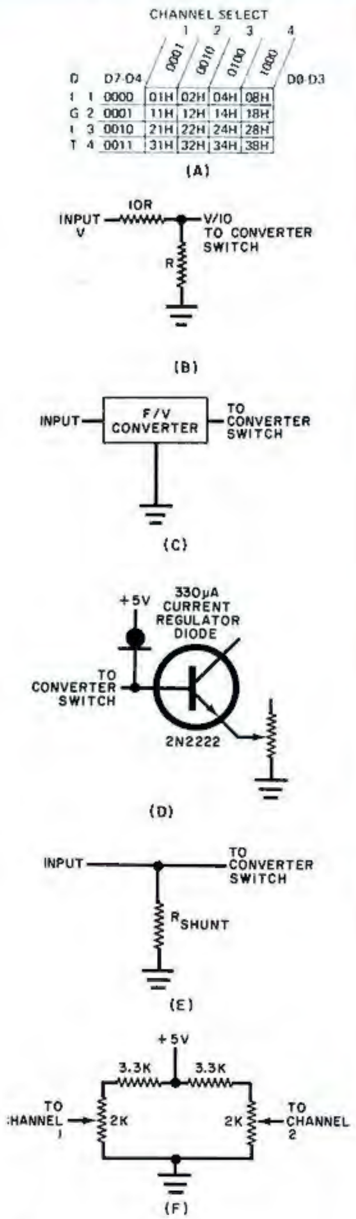


Fig. 1. Analog switch IC2 selects the input drive for A/D converter IC1. Up to four inputs can be used.

PARTS LIST

C1—100-pF disc
C2, C3—0.22-μF, 10-volt Mylar
C4—10-μF, 10-volt electrolytic
IC1—3511 A/D converter (National)
IC2—4066 quad analog switch
R1—5000-ohm, 10-turn trimmer pot.
R2—6800-ohm, ¼-watt resistor
R3—470-ohm, ¼-watt resistor
R4, R5—100,000 ohm, ¼-watt resistor
R6 through R9—1000-ohm, ¼-watt resistor
Misc.—Printed circuit board; edge connector; multilead ribbon connector; IC sockets (optional); hookup wire; solder; etc.
Note—The following is available from Alpha Electronics, Box 1005, Merritt Island, FL 32952 (tel. 305-453-3534): complete kit of parts, excluding wire and sockets, No. A/D4, for $29.95 plus 3.50 postage and handling. Also available separately: pc board No. PC178 for $9.00 + 1.00 p&h; and IC1, No. 3511, for $15.00 + 1.00 p&h.



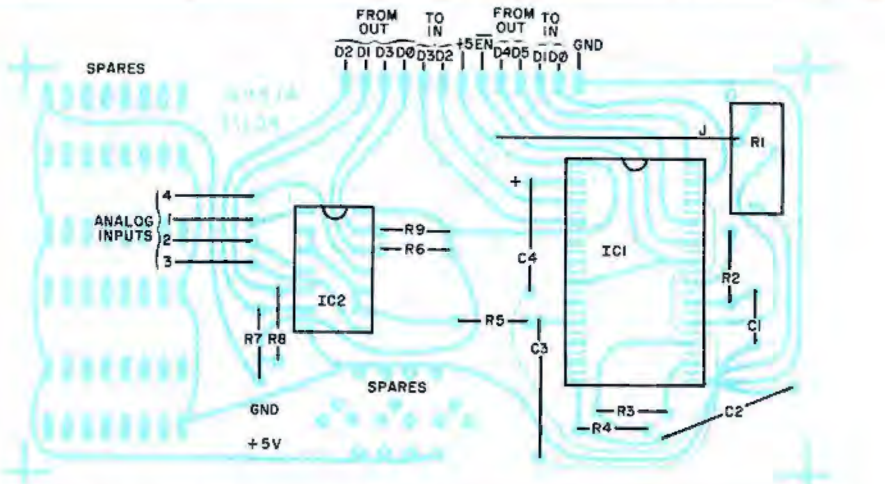


Fig. 3. Actual-size foil pattern (below) and component layout (above).

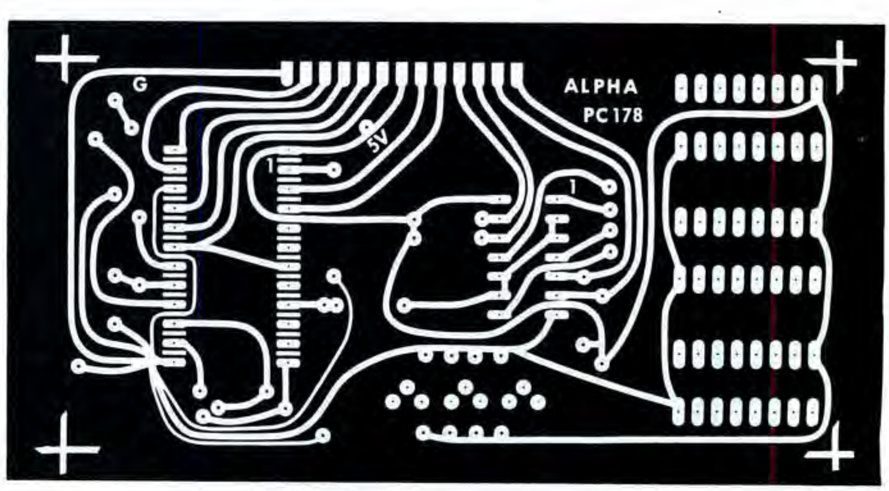


Fig. 2. Channel-digit select is shown at (A); a 10:1 voltage divider is at (B); a frequency-to-voltage scheme at (C); temperature converter (D); current measurement (E); and joystick input (F).

TABLE I—8080 ASSEMBLY LISTING

Assembly listing for 8080 (IMSAI). Inputs three most significant digits and writes to front panel.

4000	3E 11	BGN	MVI	A, 11H	Load A with Dig2, Ch1
4002	CD 38 40		CALL	INPUT	
4005	32 35 40		STA	DIG2	
4008	3E 21		MVI	A, 21H	Dig3, Ch1
400A	CD 38 40		CALL	INPUT	
400D	32 36 40		STA	DIG3	
4010	3E 31		MVI	A, 31H	Dig4, Ch1
4012	CD 38 40		CALL	INPUT	
4015	32 37 40		STA	DIG4	
4018	3E 00		MVI	A, 00H	
401A	2A 35 40		LHLD	DIG2	Dig2 L, Dig3 H
401D	BC		CMP	H	Compare H for 0
401E	06 00		MVI	B, 00H	Clear B
4020	C4 4A 40		CNZ	SUB	Gosub if H≠0
4023	3A 37 40		LDA	DIG4	
4026	FE 00		CPI	00H	
4028	CA 2D 40		JZ	WRT	If A≠0, fall thru
402B	3E 64		MVI	A, 64H	
402D	80	WRT	ADD	B	A=A+B
402E	2F		CMA		Invert data
402F	D3 FF		OUT	0FFH	Write it
4031	C3 00 40		JMP	BGN	Do again
4034		STR	DS	01	
4035		DIG2	DS	01	
4036		DIG3	DS	01	
4037		DIG4	DS	01	
4038	D310	INPUT	OUT	10H	Setup port 10H
403A	CD 40 40		CALL	DLY	A/D settling time
403D	DB 10		IN	10H	Input A/D data
403F	C9		RET		
4040	11 00 30	DLY	LXI	D, 3000H	200-ms delay
4043	1B	UP	DCX	D	
4044	7A		MOV	A, D	
4045	B3		ORA	E	
4046	C2 43 40		JNZ	UP	If D>0 do again
4049	C9		RET		
404A	C6 0A	SUB	ADI	0AH	A+A+0AH
404C	25		DCR	H	H-H-1
404D	C2 4A 40		JNZ	SUB	If H>0, do again
4050	85		ADD	L	A=A+L
5051	47		MOV	B, A	B=A
4052	C9		RET		
4053	76		HLT		

TABLE II—6800 OP CODE LISTING

Op code listing for 6800. Inputs four digits from channel 1 and stores data in memory in BCD format.

01	BDN	LDAA	01H	Load digit 1, channel 1 into A
02		BSR	INPUT	
03		STAA	DIG1	Store A in memory
04		LDAA	11H	
05		BSR	INPUT	
06		STAA	DIG2	
07		LDAA	21H	
08		BSR	INPUT	
09		STAA	DIG3	
10		LDAA	31H	
11		BSR	INPUT	
12		STAA	DIG4	
13		JMP	BGN	Do again
14	DIG1	RES	01	Res —reserves one byte of memory
15	DIG2	RES	01	
16	DIG3	RES	01	
17	DIG4	RES	01	
18	INPUT	STAA	insert	output port #
19		BSR	DLY	200—ms delay for A/D settling
20		LDAA	insert	input port #
21		RTS		
22	DLY	LDAA	--	enter values here and 23 to create a 200—ms delay
23	UP	LDAB	–	
24	UP1	SUBB	01	
25		BGT	UP1	do until B=0
26		SUBA	01	
27		BGT	UP	do until A=0
28		RTS		

TABLE III—ASSEMBLY LISTING FOR 2650

Assembly listing for 2650 (Central Data). Inputs three most significant digits and writes to data bus on WRTD instruction. Converts to hex before outputting data.

1600	75 FF		CPSL	FF	Clear PSL and setup register bank 0
1602	05 11	BGN	LODI, 1	11	Setup for channel 1, digit 2
1604	3B 20		BSTR, 3	INPUT	
1606	CA 1B		STRR, 2	DIG2	
1608	05 21		LODI, 1	21	
160A	3B 1A		BSTR, 3	INPUT	
160C	CA 16		STRR, 2	DIG3	
160E	05 31		LODI, 1	31	
1610	3B 14		BSTR, 3	INPUT	
1612	CA 11		STRR, 2	DIG4	
1614	08 0D		LODR, 0	DIG2	
1616	0B 0C		LODR, 3	DIG3	
1618	BB 24		BSFR, 0	SUB	BR if DIG3≠0
161A	0B 09		LODR, 3	DIG4	
161C	18 02		BCTR, 0	WRT	BR if DIG4=0
161E	84 64		ADDI, 0	64	
1620	F0	WRT	WRTD, 0		Write it
1621	1B 5F		BCTR, 3	BGN	do again
1623	00	DIG2	RES	01	RES 01 reserve one byte
1624	00	DIG3	RES	01	
1625	00	DIG4	RES	01	
1626	D5 00	INPUT	WRTE, 1	00	Setup port 00
1628	3B 03		BSTR, 3	DLY	A/D settling time delay
162A	56 00		REDE, 2	00	read port 00 into R2
162C	17		RETC, 3		
162D	77 10	DLY	PPSL	10	select register bank 01
162F	05 FF		LODI, 1	FF	gives 200 ms delay with 1633
1631	A5 01	UP	SUBI, 1	01	
1633	06 40		LODI, 2	40	
1635	A6 01	UPI	SUBI, 2	01	
1637	5A 7C		BRNR, 2	UP1	do again until R2=0
1639	59 76		BRNR, 1	UP	do until R1=0
163B	75 10		CPSL	10	select register bank 00
163D	17		RETC, 3		
163E	A7 01	SUB	SUBI, 3	01	R3=R3-1, converts BCD to hex
1640	84 0A		ADDI, 0	0A	R0=R0+0A
1642	5B 7A		BRNR, 3	SUB	do until R3=0
1644	17		RETC, 3		
1645	40		HALT		

(Continued from page 45)

mines the reference voltage used by *IC1*. The other passive components determine clock frequency, provide signal filtering, and interconnect *IC1*.

Software. The digit and channel select codes are shown in Fig. 2A. The values shown are in hexadecimal code. To use the table, move down the rows until the proper digit is located. Then move over until the proper channel is located. The hex number at this point is the data to be written to the output port to set up the converter. The strobe that enables the output port (EN) must be active low when connected to the converter. If necessary, an inverter can be wired into the circuit to perform the inversion.

When reading data from the converter, it is necessary to access only the correct input port. Examples of programs written for an 8080, 6800, and 2650 are shown in Tables I through III. The program flow is essentially the same for any 8-bit computer. The digit/channel information is loaded into a register and then the program is stepped to a subroutine (INPUT) and outputs that register to the output port. A 200-ms delay (DLY) subroutine is used to allow the A/D converter to settle. Then the data is read from the input port into a register.

Upon returning from the INPUT subroutine, the BCD digit is stored in memory (DIGX) and is repeated for each digit required before branching back to the

TABLE IV—BASIC SAM GAME

```
SAM (surface-to-air missile) GAME
Central Data Basic (2650)

0000          RESTORE
0010          READ, R, W, M, P, Z
0020          DATA 0, -1, 0, 11, 0   REM sets up port 0, chan#1, digit#3
0100          EXTOUT 0, 33
0105          X=SIN(1)   REM delay for A/D settling
0110          EXTIN 0, B   REM reads port 0, chan#1, digit#3 into B
0120          EXTOUT 0, 49
0125          EXTIN 0, A
0130          EXTOUT 0, 34
0135          X=SIN(1)
0140          EXTIN 0, D
0150          EXTOUT 0, 50
0155          EXTIN 0, C
0160          A=INT (A*10+B)*.8)+1
0165          C=(C*10+D)*4
0170          ERASE   REM clears screen
0175          IF R>17 GOTO 1010
0200          W=W+1
0210          P=P-1
0220          IF W>10 W=10
0230          IF P<0 P=0
0240          PRINT@14, 15' MISSILES FIRED'#W
0250          PRINT@15, 15' MISSILES LEFT'#P
0300          PRINT@13, 9'I'
0310          PRINT@14, 8'I'
0320          PRINT@15, 5'IIIII'
0330          PRINT@10, 50'I I I'
0340          PRINT@11, 50'IIIIIII'
0400          Z=Z+4
0410          Y=1+INT(RND(7))
0500          PRINT@Y, Z'++++++'
0510          PRINT@Y-1, Z'    +'
0520          PRINT@Y+1, Z'    +'
0530          IF Z>50 GOTO 0900
0600          READ Q, V, L
0610          DATA 12, 9, 1
0620          RESTORE 0610
0630          IF W=0 GOTO 0800
0640          PRINT@Q, V'↑'
0650          IF C+1>19 L=3
0660          IF C+1>31 L=6
0670          IF V>C+1 V=V-1
0680          IF V<C+1 V=V+L
0700          Q=Q-1
0710          IF Q=A IF V=C+1 GOTO 0740
0720          IF Q<A GOTO 0740
0730          GOTO 0640
0740          PRINT@A, C+1'X'
0800          PRINT@14, 40'TARGET: RANGE'#Y' BEARING'#Z
0810          PRINT@15, 40'MISSILE: RANGE'#A' BEARING'#C+1
0820          IF C>Z IF C<Z+6 GOTO 0880
0825          IF P=0 M=1
0830          IF M=1 GOTO 0170
0835          IF R>0 GOTO 0850
0840          INPUT'    FIRE'R
0845          IF R=22 GOTO 0100
0850          R=R-1
0860          IF R<0 GOTO 0800
0870          GOTO 0100
0880          PRINT@A.C+1'X DESTROYED'
0890          PRINT@15, 5
0895          GOTO 1000
0900          PRINT@11, 50'DESTROYED'
0910          PRINT@15, 5
1000          STOP
1010          PRINT@A, C+1'X'
1020          PRINT@15, 40'MISSILE:   RANGE'#A'   BEARING'#C+1
1030          R=R-1
1040          IF R>17 GOTO 0100
1050          R=0
1060          GOTO 0840
1070          END
```

This program prints a missile launching site, a factory, an airplane (bomber), and a printout of the airplane's and missile's range and bearing. When FIRE appears at the bottom of the screen, type a number (1 through 10) for the number of missiles you wish to fire. The missiles will fire in sequence. Type 22 to clear the screen and display the missile range and bearing adjustments. You may then adjust the controls to alter these values. After 5 shots, the program will return to FIRE. If you input 1 (CRLF), then carriage-return/line-feed, an arrow will print the track of the missile until it reaches its range, and an X will appear to simulate an explosion. The object is to hit the plane on its fuselage, in which case, X DESTROYED will be printed. You have 10 missiles. If you do not destroy the plane in 10 shots, or if the plane reaches space 50 on the screen before you destroy it, the plane will destroy the factory. The aircraft will progress across the screen at the rate of 4 spaces for each missile fired. However, the plane will move up and down by a random amount (line 0410 controls this).

Note that the FIRE 22 routine does not subtract from the missiles remaining, but you cannot destroy the plane in this routine either. All entries must be followed by CRLF. After each missile is fired, and FIRE is displayed, you can adjust the range and bearing controls to alter the missiles course and range.

beginning (BGN). The 8080 listing also includes a routine to convert the three most significant digits into hex code and place the result in storage (STR). When programming, allow for the fact that this data has no decimal point.

Some typical input adapters are shown in Fig. 2B through 2F; B, C, and E are conventional, while D illustrates a temperature converter. If you use this or a similar circuit, allow for any voltage offset in this type of converter. Also, keep in mind that only the two least significant digits are required for a temperature reading. This data should be viewed as relative and not absolute. Decisions should be based on exceeding a relative number, rather than a specific number of degrees. For example, if the temperature converter is adjusted for an output of 1.050 volts at 25° C and the voltage decreases by 2.3 mV/°C, the program can be written to do something when the temperature is 20° C, or 1.039 volts. The 1.039 volts is related to the temperature but is not actually in degrees.

Figure 2F illustrates a joystick (or two independent potentiometers) for use in game programs.

A BASIC program to play the game SAM is shown in Table IV. Note that the data written to the output port (EXTOUT) is in decimal, rather than in hex. The REM statements should help explain the program. Table V illustrates a 4-channel DVM program, which is also written in BASIC.

Construction. An actual-size etching and drilling guide and a component-installation diagram for the A/D converter are shown in Fig. 3. During assembly, note the polarity of *C4* and, if you wish, use sockets for the IC's. Note also that there are provisions on-board for optional inverters (IC's or discrete transistors); these can be Wire Wrapped.

When installing the IC's, observe the usual precautions for handling MOS devices. Since the 5-volt power supply is also used as the reference, make sure it is well regulated and stable. After assembly, adjust *R1* for as near to an exact 2.000 volts at pin 16 as possible.

If your system employs an active high strobe, use an inverter. Flat ribbon cable can be used to interconnect the converter to the host computer. If desired, a 16-pin socket can be mounted at one of the extra positions on the board, and, with Wire Wrap, it can be used to make the external connections instead of the edge connector shown.

Testing. After assembling the board

TABLE V—4-CHANNEL DVM PROGRAM

```
4-Channel DVM Program
Central Data Basic (2650), Version 1.2

000        RESTORE
010        DATA 1, 17, 33, 49, 1, 2, 18, 34, 50, 2, 4, 20, 36, 52, 3, 8, 24, 40, 56, 4
020        READ A, B, C, D, E        REM A=Ch=Digit 1:B=Ch=Digit 2:C=Ch=
                                         Digit 3:D=Ch=Digit 4:E=Ch=
030        GOSUB 100
040        IF E=4 GOTO 000
050        GOTO 010
100        EXTOUT 0, A               REM sets up I/O port Ch&digit
110        S=SIN(1)                  REM delay for A/D settling time
120        EXTIN O, Z                REM reads port 0 into Z
130        EXTOUT 0, B
140        EXTIN 0, Y
150        EXTOUT 0, C
160        EXTIN 0, X
170        EXTOUT 0, D
180        EXTIN 0, W
190        PRINT@E, 5'CHANNEL#'=E'='=(W*1000+X*100+Y*10+Z)' MILLIVOLTS'
                                     REM prints at line E, character position 5,
                                         channel #E=(voltage) MILLIVOLTS
200        RETURN
300        STOP
310        END
```

and adjusting *R1* for the 2-volt reference, load a driver program and check the system for accuracy. If the data appears to be unstable, check the 200-ms delay between the output port strobe and the input port strobe. You may have to vary the values loaded into the DLY routine until the correct delay is obtained. This delay is required only when the channel information is first changed. If only one channel is used, the delay need be used only the first time the channel is selected.

The 6800 program shown is not exact; it is given here as an example of a driver routine for the 6800 CPU. The 8080 program, tested in an 8080-based computer, uses I/O port 10H and a delay (DLY) routine for a 2-MHz system. The CMA and OUT-OFFH instruction invert the data and write it into the front panel. Location 4034 reserves one byte of memory that can be used to store the hex data, if desired. The 2650 program was written for and tested on a Central Data computer. The port used here was OOH, and the DLY routine is for a 1.25-MHz clock.

The A/D converter offers the computer user an inexpensive way for his computer to "communicate" with the analog happenings that dominate our lives. It offers a multitude of ways of sensing and measuring analog data not possible in a simple keyboard-entry system. ◇