# Popular Electronics®

WORLD'S LARGEST SELLING ELECTRONICS MAGAZINE
JANUARY 1982/$1

HOW TO ADD

- Safe, Convenient Shutoff to Smoke Detectors
- "Real World" Signal Handling to TRS-80 Computers
- Overseas Broadcast Reception to Any AM Radio

## $70 Decoder for New CX Records

Enjoy 20-dB Noise Reduction From In-The-Groove Encoding




Cassette Recorder

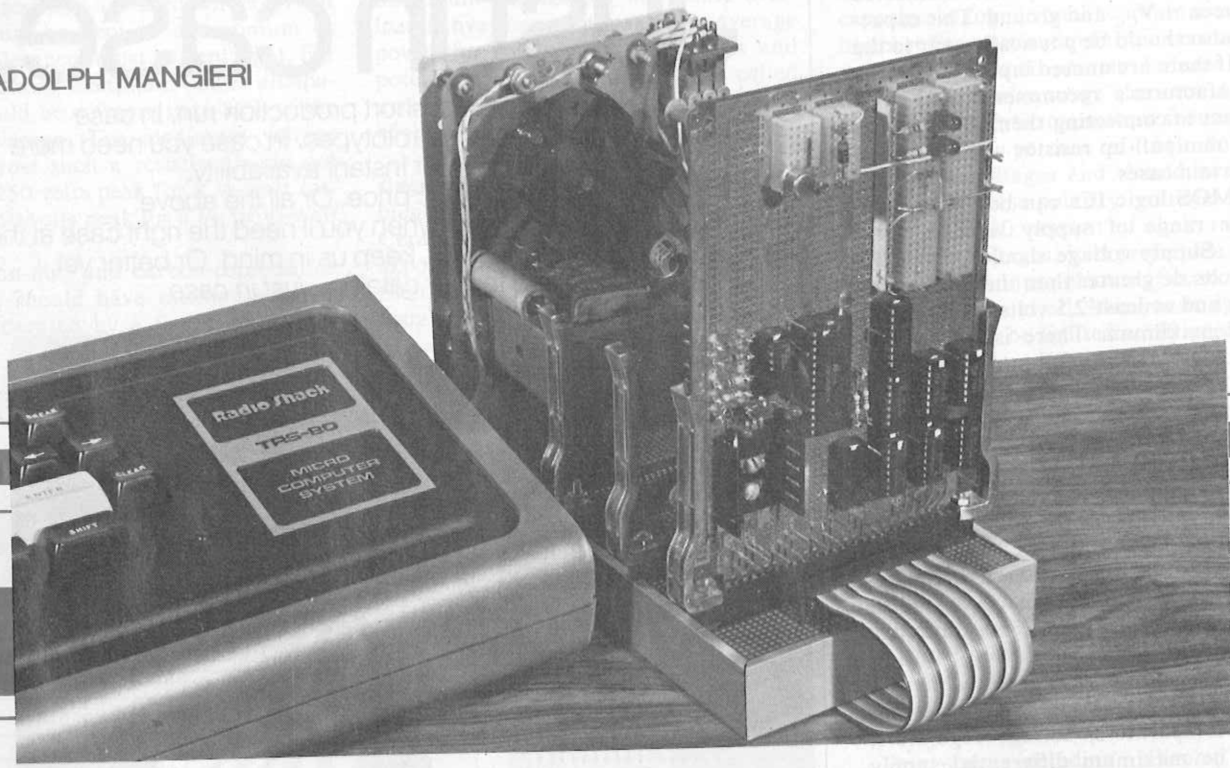
e

EPI Model A300 Speaker

# ANALOG-DIGITAL CONVERTER FOR TRS-80 INTERFACING

*An 8-bit, 8-channel digital circuit that allows you to connect analog voltages to your TRS-80 microcomputer*

BY ADOLPH MANGIERI




DIGITAL computers "speak and understand" only the binary language of electrical ones and zeros. Unfortunately, the binary language is not suitable for direct measuring of physical quantities such as voltage, pressure, temperature, light, or other continuously varying (analog) parameters. To utilize the digital computer in measurement and control systems, an analog-to-digital interface is required. Fortunately, such analog-to-digital converters (ADC) are now available at low cost.

Interfacing with the TRS-80 Model I microcomputer, the 8-bit, 8-channel ADC covered in this article includes a four-bit output port for controlling lamps, relays, and other devices. The output port is readily expandable to eight channels, thus providing 32 channels of control. Running in the TRS-80 TBUG monitor, the accompanying machine language program ANADIG shows how to structure a multichannel data-acquisition system. Several input and output circuits are detailed, including means to quantize the range of an input channel to output multiple decisions controlling a number of output circuit branches. The ADC accepts an input voltage and converts it to binary form for display or further processing by a computer. Common converter types include the costly high-speed 'flash' converter, the inexpensive but slow ramp converter, and the successive-approximation converter that provides excellent speed at relatively low cost. In all cases, the ADC seeks to match the level of an analog input signal with stepped and weighted reference voltages and generate a binary value when the match is found.

Considering first the successive approximation converter, Table I shows conversion of input signal of weight 67 in eight approximations taken in sequence. On the first comparison (bit D7), weight 128 is greater than 67 thus it is discarded by setting output bit D7 to zero. On trial two, weight 64 is less than 67 and is retained as a partial sum by setting bit D6 to one. The following comparisons through bit D2 are discarded because the partial sum would exceed 67. The remaining two trials bring the sum to exactly 67 and the corresponding data bits are set to one causing 67 to be converted to 01000011 or 43 hex. For an input signal of weight 255, the data bits are set to 11111111 yielding FF hex or full-scale. The converter resolves full-scale input of one part in 256.

The block diagram of Fig. 1 shows the internal circuit blocks of the eight-chan-

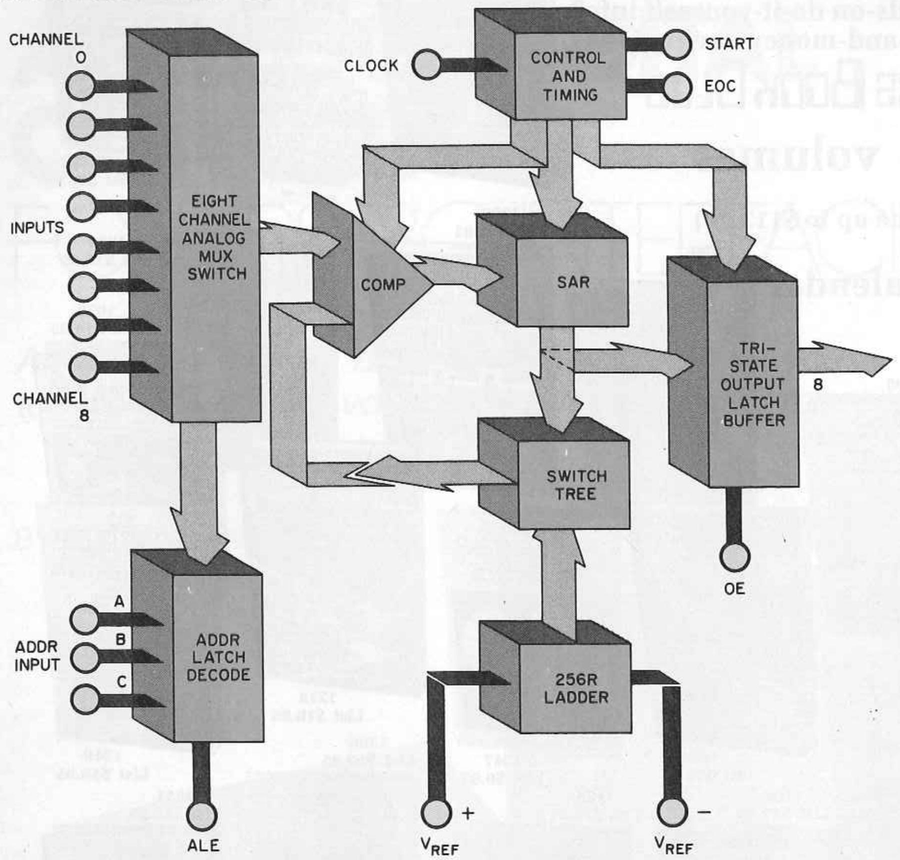



*Fig. 1. Block diagram of the circuits contained in the AD0809 analog-to-digital converter used in the project.*

nel converter used in this project. One of eight input signals is applied to one input of the comparator through a mux (multiplex) analog switch. The particular channel selected depends on the address bits applied to the address data latch decoder. Bit 000 selects input channel 0, bit 001 selects channel 1, and so forth up to channel 7 by bit 111.

A stable 5-volt reference is applied to a 256R resistor ladder network that supplies weighted reference voltages for comparisons. With the input signal present at one input of the comparator, a switch tree sequentially selects and applies weighted reference voltages to the other comparator input. The comparator output feeds into the successive approximation register (SAR) which performs logical decisions and assembles the binary output data in the Tri-State data-out latch and buffer. The ADC clock and timing and control circuits determine the sequence of events. Using this arrangement, at a clock frequency of 640 kHz, conversion takes place in 116 microseconds.

Figure 2 shows the ADC timing. With address bits and input signal present, address latch enable pulse (ALE) strobes the address bits into the address latch decode circuit. Pulse START initiates conversion and end-of-conversion pulse (EOC) goes low during conversion. Following conversion, pulse EOC goes high and pulse output enable (OE) is applied to enable the data onto the bus for acceptance by the computer.

For comparison, a six-bit parallel flash converter includes a resistor ladder supplying 63 reference voltages each connected to one input of 63 comparators. The input signal connects to the other input of all comparators. Comparisons take place all at once thus the name "flash" converter. The 63 outputs of the comparator string are then decoded by extensive and complex logic to form the equivalent binary output. However, an eight-bit flash converter requires 255 comparators! Costly to manufacture, the flash converter is usually limited to six bits or less.

**Table I**

**EXAMPLE OF SUCCESSIVE APPROXIMATION**

| Bit | Weight | Comparison | Bit | Sum |
|---|---|---|---|---|
| D7 | 128 | 128>67 | 0 | 0 |
| D6 | 64 | 64<67 | 1 | 64 |
| D5 | 32 | 32+64>67 | 0 | 64 |
| D4 | 16 | 16+64>67 | 0 | 64 |
| D3 | 8 | 8+64>67 | 0 | 64 |
| D2 | 4 | 4+64>67 | 0 | 64 |
| D1 | 2 | 2+64<67 | 1 | 66 |
| D0 | 1 | 1+66=67 | 1 | 67 |

The ramp ADC technique uses a digital-to-analog converter (DAC) and a computer program to generate a staircase voltage ramp of 256 steps for use by the eight-bit converter. The stepped output of the DAC and the input signal connect to comparator inputs, and on each successive voltage step, the computer program checks comparator output and advances to the next step if the match is not found. Two hundred and fifty-five comparisons are required to reach full-scale for eight-bit conversions. Though relatively slow, ramp-conversion techniques offer advantages through software control.

**Circuit Operation.** As shown in the schematic of Fig. 3, clock generator *IC8* is a 555 timer operating at approximately 100 kHz. Input channels 0 and 1 are the only ones used at this time, with the remaining 6 input channels grounded. A zener diode (*D2, D3*) and a capacitor (C5, C6) protect the active CMOS input channels.

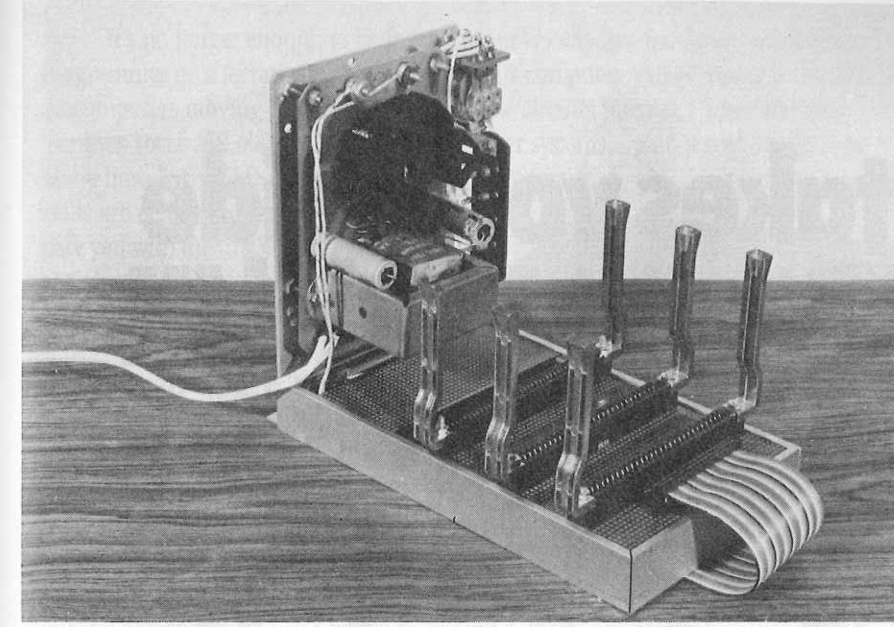
When the program ANADIG issues an $\overline{\text{OUT}}$ instruction to port address $\overline{\text{FD}}$ (decoded by *IC5*), the instruction transmits channel address bits on data lines D0, D1, and D2. Thus *IC5* in conjunction with the $\overline{\text{OUT}}$ signal activates *IC6A* pulsing ALE and START inputs with the address latched into the ADC. A time delay in the program allows time for completion of the conversion. The program then issues an $\overline{\text{IN}}$ instruction to port address $\overline{\text{FD}}$ causing *IC6B* to activate OE (output enable) and placing the converted data on the data bus as input to the computer.

A program task subroutine then processes the data and makes a decision for use by output port, *IC7*. The task decision is output to port address $\overline{\text{FB}}$ decoded by gate *IC4* and enabling *IC7*. Data bits D0 through D3 are transmitted to *IC7* and determine the output of four data latches used to control external indicator lamps or relays.

In the case of an external transistor driver (*Q1*), zener diode *D4* protects the circuit in the event of failure of the transistor. Voltage regulator *IC11* supplies five volts to the circuit.

Additional input channels may be connected to *IC9* as required. Additional output channels are created by adding another 74LS75 (*IC7*) and connecting device pins 4 and 13 to pin 10 of *IC6C*. Data lines D4 through D7 are passed through the spare buffers of *IC3* to the inputs of the second data latch.





*The system is assembled on an aluminum frame with power supply and card guides.*

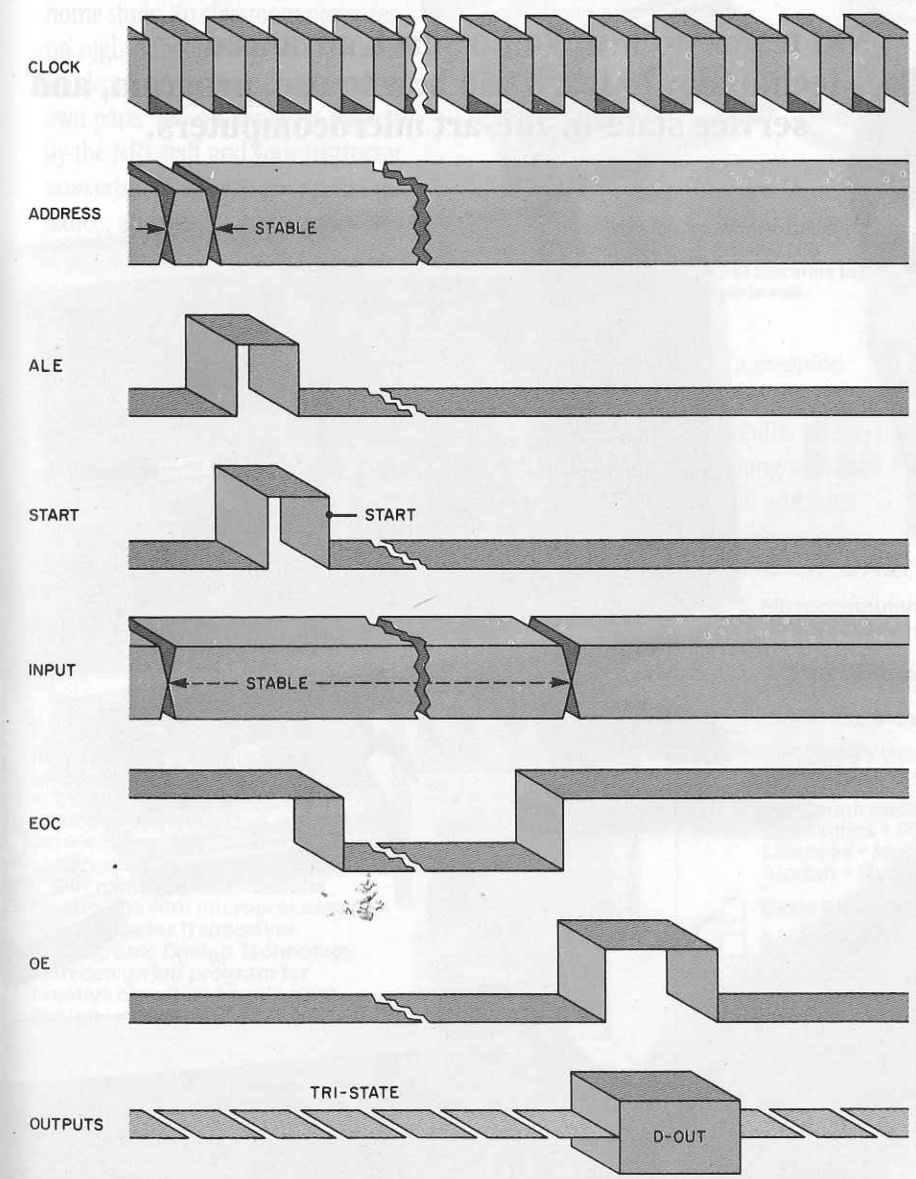



*Fig. 2. Timing diagram of the analog-to-digital converter IC.*

**Construction.** The circuit was assembled on a Vector 4494 ANY-DIP plug card and wire wrapped. Install *IC1, IC2, IC3*, and *IC10* in the socket row near the card fingers. Install *IC4, IC5*, and *IC6* in the second socket row and place *IC7, IC8*, and *IC9* in the third row. Experimentation with assorted input and output circuits can be facilitated by installing pairs of Vector T66-96 and T66-32 Klip Bloks and two T45-24 Klip Buses on the upper portion of the card as shown. The plug-board system shown consists of three Vector R644-3 44-contact card receptacles and three pairs of BR27D card guides installed on the 51X aluminum frame. Install two rows of T46-5-9 board pins with pin faces aligned on perfboard at one end forming a male IDC connector. Use a 40-line IDC cable disconnect at the plug board system. A Jameco No. JE210 5-to-15-volt adjustable regulated power supply set for 12-volts powers the circuit. Do not run the TRS-80 5-volt supply to the plug board system. As an alternative to the plug board system, assemble the circuit on Vector 8002 or 8004 Circbords for wire wrapping and install the card in the 51X aluminum frame.

**Checkout.** With integrated circuits removed and ribbon cable disconnected from the computer, power the voltage regulator and check voltage distribution at the pertinent socket pins. Check voltages at the far end of the cable and be certain that supply voltages do not feed back to the computer. Install the integrated circuits taking usual precautions when handling the CMOS converter chip. Energize the circuit and verify presence of clock pulses at pin 3 of *IC8* using either a counter or oscilloscope. With power off, make connections to the computer and verify proper operation of the computer. Look for shorted bus lines if the computer fails to function.

Connect the input test circuit shown on the schematic diagram to the input of channel zero. Jumper channel-one input to channel-zero input. Enter and load program ANADIG into memory using the TRS-80 T-BUG monitor and break the looping program by inserting STOP code CD 91 40 at address 4A27H. Set test potentiometer *R6* to its ground end, and run the program. Both *LED1* and *LED2* should turn off. Verify that data input buffer memory location 4A00H and 4A01H hold data 00 and that output buffer 4A08H holds 00. Set the test potentiometer to five volts and observe that both LEDs glow. Verify that chan-

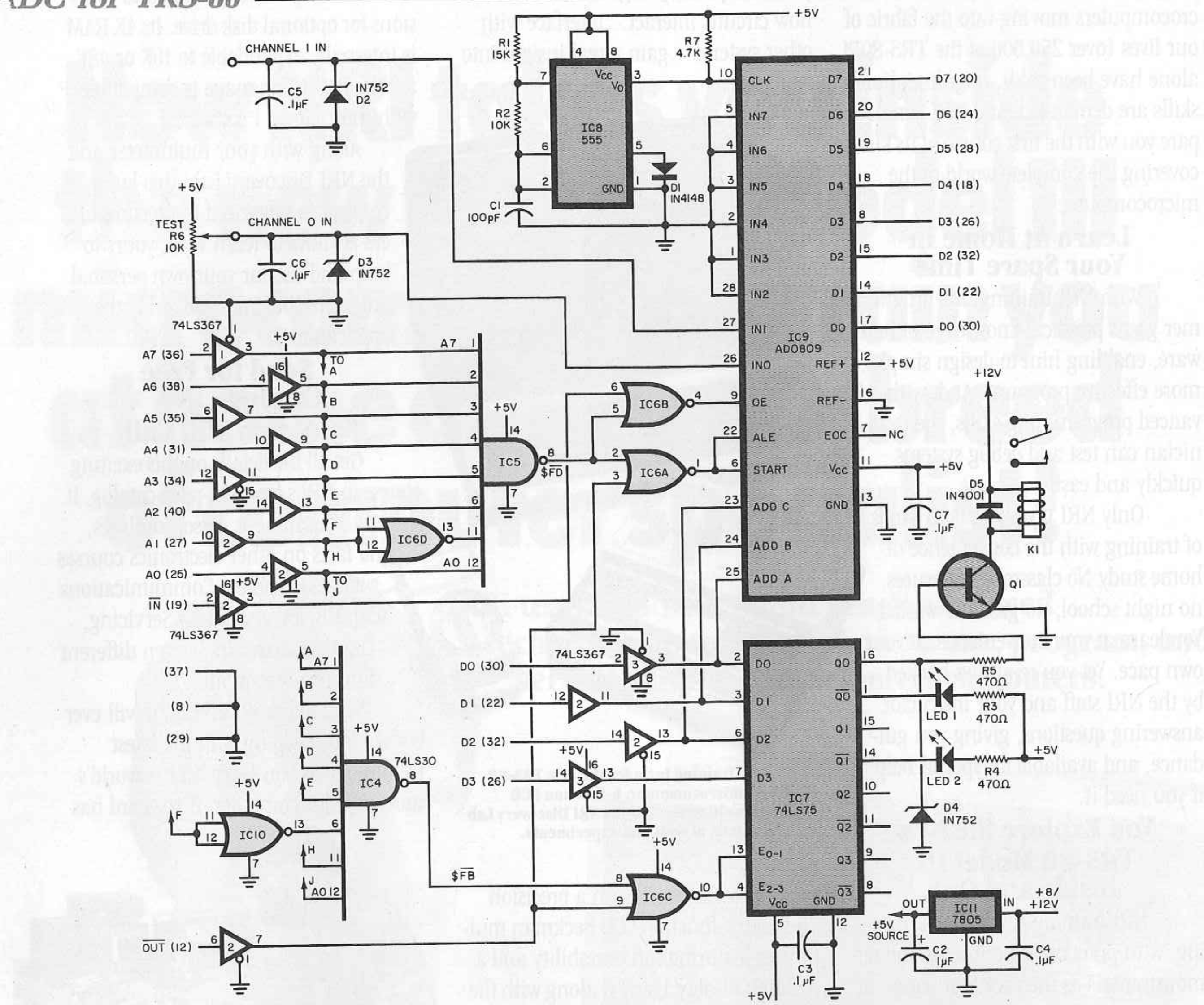



*Fig. 3. A 555 clock generator, IC8, operates at approximately 100 kHz. Input channels 0 and 1 are the only ones used in this application.*

## PARTS LIST

C1—100-pF ceramic capacitor
C2—1.0-$\mu$F, 35-V tantalum electrolytic
C3 through C7—0.1-$\mu$F, 15-V disc capacitor
D1—1N4148 switching diode
D2,D3,D4—1N752 5.6-V zener diode
D5—1N4001 rectifier
IC1,IC2,IC3—74LS367 three-state hex buffers
IC4,IC5—74LS30 8-input NAND gate
IC6,IC10—74LS02 quad 2-input NOR gate
IC7—74LS75 4-bit data latch
IC8—555 timer
IC9—ADC 0809 eight-bit, eight-channel ADC (available from Jameco Electronics, 1355 Shoreway Rd., Belmont, CA 94002)
IC11—7805 5-V, 1-A voltage regulator
K1—12-V relay
LED1,LED2—Light-emitting diode (XC-526R or equiv.)
Q1—2N3053, RS-276-2030 npn transistor
R1—15-kΩ, ¼-W resistor
R2—10-kΩ, ¼-W resistor
R3,R4,R5—470-Ω, ¼-W resistor
R6—10-kΩ potentiometer
R7—4.7kΩ, ¼-W resistor
Misc: Vector 4494 plug board; 51X aluminum frame; R644-3 44-contact card receptacles (3); BR27D card guides (6); T46-5-9 board pins; perfboard; 28-pin DIP socket; 16-pin DIP sockets (4); 14-pin DIP sockets (4); T49 Klip-Wrap posts; T66-96, T66-32 Klip Bloks (pairs); T45-24 Klip Bus (2); ribbon cable and connectors; wire; etc.

nels zero and one data buffers hold data FF or 255 while outport buffer 4A08H now holds 03. Vary *R6* slowly about the trip points. Notice that LED1 flickers and relay *K1* chatters at the trip point. Notice that the turn-on and turn-off points of *LED2* differ slightly and with no flicker. This is the result of deadband or hysteresis built into the task program of channel one.

**Software.** Use program ANADIG shown here for initial experiments with multichannel data acquisition and processing and write your own programs for specific applications. The looping main program RUN performs initializations, issues channel addresses, and CALLS subroutine START which initiates A/D conversions followed by subroutines TASK0 and TASK1 which control the relay outport. Memory locations 4A00H through 4A07H respectively store input data for channels zero through seven. Location 4A08H holds the outport byte common to all output channels.

Tracing through a typical program run, index register IX is initialized to point to the input data buffer. The outport byte is arbitrarily cleared to 00. Channel zero address 00 is first loaded





## ANADIG TEST PROGRAM

```
                00100 ;FILENAME - ANADIG
                00110 ;BY ADOLPH A. MANGIERI 4/81
                00120 ;CHAN BUFFER - 4A00H THRU 4A07H
                00130 ;OUTPORT BUFFER - 4A08H
4A00            00140         ORG   4A00H
0009            00150         DEFS  9           ;BUFFERS
4A09 DD21004A   00160 RUN     LD    IX,4A00H    ;POINTER
4A0D DD360800   00170         LD    (IX+8H),00H ;CLR BUFFER
4A11 3E00       00180 LOOP1   LD    A,00H       ;CHAN 0
4A13 CD7A4A     00190         CALL  START       ;START A/D
4A16 DD7700     00200         LD    (IX+0),A    ;SAVE DATA
4A19 CD9D4A     00210         CALL  TASK0       ;DO TASK 0
4A1C 3E01       00220         LD    A,01H       ;CHAN 1
4A1E CD7A4A     00230         CALL  START       ;START A/D
4A21 DD7701     00240         LD    (IX+1),A    ;SAVE DATA
4A24 CDDB4A     00250         CALL  TASK1       ;DO TASK 1
4A27 18E8       00260         JR    LOOP1       ;LOOP
0051            00270         DEFS  81          ;SPACE
4A7A C5         00280 START   PUSH  BC          ;SAVE
4A7B D3FD       00290         OUT   (0FDH),A    ;START A/D
4A7D 062F       00300         LD    B,2FH       ;TIME DELAY
4A7F 10FE       00310 LOOP2   DJNZ  LOOP2       ;LOOP2
4A81 DBFD       00320         IN    A,(0FDH)    ;GET DATA
4A83 C1         00330         POP   BC          ;RESTORE
4A84 C9         00340         RET               ;RETURN
0018            00350         DEFS  24          ;SPACE
4A9D F5         00360 TASK0   PUSH  AF          ;SAVE
4A9E E5         00370         PUSH  HL          ;SAVE
4A9F D5         00380         PUSH  DE          ;SAVE
4AA0 2600       00390         LD    H,00H       ;CLEAR H
4AA2 1600       00400         LD    D,00H       ;CLEAR D
4AA4 2E7C       00410         LD    L,7CH       ;TRIP POINT
4AA6 B7         00420         OR    A           ;CLEAR CARRY
4AA7 DD5E00     00430         LD    E,(IX+0)    ;GET DATA
4AAA ED52       00440         SBC   HL,DE       ;COMPUTE
4AAC FAB54A     00450         JP    M,SET0      ;GO IF NEG
4AAF DDCB0886   00460         RES   0,(IX+8)    ;RES BIT 0
4AB3 1804       00470         JR    LDPORT      ;EXIT
4AB5 DDCB08C6   00480 SET0    SET   0,(IX+8H)   ;SET BIT 0
4AB9 DD7E08     00490 LDPORT  LD    A,(IX+8H)   ;GET DATA
4ABC D3FB       00500         OUT   (0FBH),A    ;SEND DATA
4ABE D1         00510         POP   DE          ;RESTORE
4ABF E1         00520         POP   HL          ;RESTORE
4AC0 F1         00530         POP   AF          ;RESTORE
4AC1 C9         00540         RET               ;RETURN
0019            00550         DEFS  25          ;SPACE
4ADB F5         00560 TASK1   PUSH  AF          ;SAVE
4ADC E5         00570         PUSH  HL          ;SAVE
4ADD D5         00580         PUSH  DE          ;SAVE
4ADE 2600       00590         LD    H,00H       ;CLEAR H
4AE0 1600       00600         LD    D,00H       ;CLEAR D
4AE2 2E7E       00610         LD    L,7EH       ;HI LIMIT
4AE4 DD5E01     00620         LD    E,(IX+1H)   ;GET DATA
4AE7 B7         00630         OR    A           ;CLEAR CARRY
4AE8 ED52       00640         SBC   HL,DE       ;COMPUTE
4AEA FAF64A     00650         JP    M,SET1      ;SET BIT 1
4AED 2E7A       00660         LD    L,7AH       ;LO LIMIT
4AEF ED52       00670         SBC   HL,DE       ;COMPUTE
4AF1 F2FC4A     00680         JP    P,RES1      ;JP IF POS
4AF4 180F       00690         JR    EXIT        ;TO EXIT
4AF6 DDCB08CE   00700 SET1    SET   1,(IX+8H)   ;SET BIT 1
4AFA 1804       00710         JR    OUTPRT      ;
4AFC DDCB088E   00720 RES1    RES   1,(IX+8H)   ;RES BIT 1
4B00 DD7E08     00730 OUTPRT  LD    A,(IX+8H)   ;GET DATA
4B03 D3FB       00740         OUT   (0FBH),A    ;SEND DATA
4B05 D1         00750 EXIT    POP   DE          ;RESTORE
4B06 E1         00760         POP   HL          ;RESTORE
4B07 F1         00770         POP   AF          ;RESTORE
4B08 C9         00780         RET               ;RETURN
0000            00790         END
0000  TOTAL ERRORS

OUTPRT   4B00
EXIT     4B05
RES1     4AFC
SET1     4AF6
LDPORT   4AB9
SET0     4AB5
LOOP2    4A7F
TASK1    4ADB
TASK0    4A9D
START    4A7A
LOOP1    4A11
RUN      4A09
```

into register A and subroutine START is CALLed. Routine START loads the address into the converter and starts A/D conversion. After a time delay set by byte 2F at address 4A7EH, the program returns to RUN with converted data in register A. Program RUN stores the data at address 4A00H and CALLS TASK0. Subroutine TASK0 fetches the stored input data and subtracts it from trip point 7C (124) located at address 4AA5H. If the result is negative, bit B0 of the outport byte is set to one or otherwise set to zero. The outport byte is then transmitted to the relay port data latch. Bit B0 now as data bit D0 may alter the status of output channel zero. The next program module of RUN addresses itself to channel one and TASK1 in a similar manner.

Subroutine TASK1 includes an upper trip point 7E (126) at address 4AE3H and a lower trip point 7A at address 4AEEH. When the converted input falls between these limits, bit B1 of the outport data byte is left unchanged. This introduces hysteresis much like a Schmitt trigger and prevents repetitive operation of mechanical relays and solenoids when the input levels hover near the trip points. The trip points and deadband are readily altered to suit the application. Use the TRS-80 TBUG machine language monitor to enter the object code. No changes are required for entry into either Level I or Level II machines. Alternatively, enter the source code or assembly listings using the TRS-80 Editor/Assembler EDTASM. Once the code is entered, make a tape copy using TBUG. Minor program changes are best entered manually using the TBUG. For major alterations, restructuring, or relocation of code, use EDTASM which markedly reduces the effort.

**Input and Output Circuits.** Input circuit Fig. 4A uses a thermistor for sensing temperature or a light-dependent resistor for sensing light levels. Resistor *Ra* can be a potentiometer for calibration or setting of trip points. The sensor and pot can be interchanged. It is best to include RC filtering in the input circuit to remove noise and ac components which affect conversion. Try 100,000 ohms for *Rb* and 0.1 $\mu$F for *Ca* or higher values if the RC time constant is not objectionable. Figure 4B shows two potentiometers of a joystick having two outputs which feed into two channels of the ADC, with the game program processing the converted data.

Low-level voltages from devices such as a photovoltaic cell or thermocouple can be amplified by an op amp such as the LM324 as shown in Fig. 4C. Stage gain or scaling depends on the ratio of

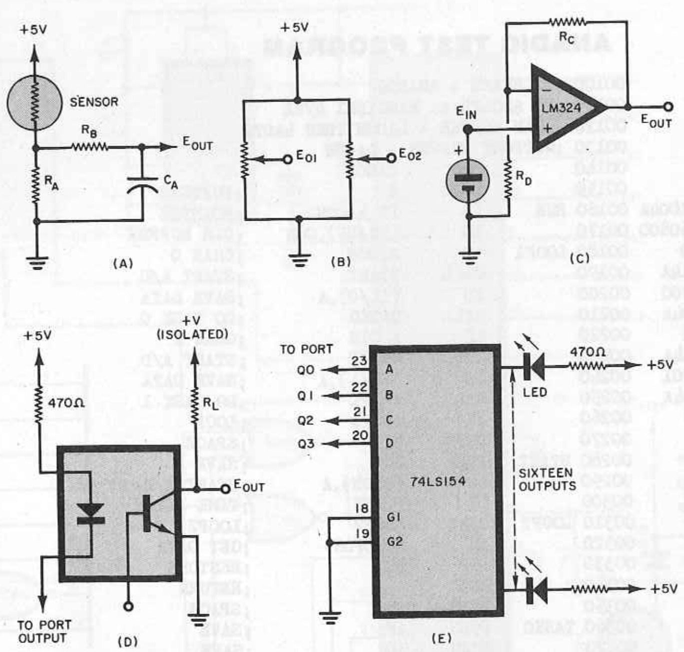



*Fig. 4. Inputs: (A) a thermistor or light sensor; (B) two outputs of a joystick; (C) an amplifier to step up low-level signals. An optoisolator (D) is used to drive a triac or SCR. A decoder can be used on the output (E) to drive a display or LED or an alarm.*

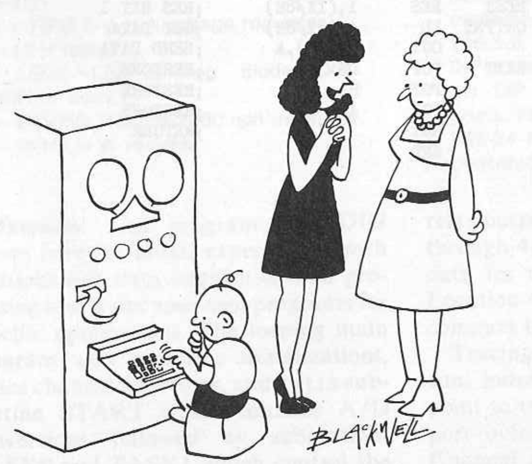
resistors *Rc* and *Rd*. For scaling, use a potentiometer for *Rc*. Higher input voltages can be scaled using a voltage divider string. For low-voltage ranges, scaling is obtained by switching suitable values of resistor *Rc*. If you use 2.55-volts dc for the ADC reference voltage, full-scale or 255 occurs with an input voltage of 2.55 volts. For video display of converted data, include a subroutine which converts binary to ASCII for display in video memory space. This and other useful subroutines such as multiplication and division can be found in manuals on Assembly language programming.

Output circuits include indicator lamps, audible alarms, mechanical and solid-state relays, stepping motors, and similar devices. Mechanical relays provide electrical isolation between the computer circuits and the controlled power circuits. Where triacs or SCRs are used to control power, use an optoisolator as shown in Fig. 4D. Alternatively, install a reed relay on the plug board for control of solid-state or remote-mounted mechanical relays.

Through program subroutines, the full-scale range of the ADC input can be divided into a number of segments each issuing a unique decision to the outport. As an example, the task program can divide full-scale range into sixteen segments with each program segment issuing a unique 4-bit nyble ranging from 0000 to 1111. The nyble is output to the 74LS75 latching port and affects the four outputs.

As shown in Fig. 4E, a 4-to-16-line decoder is connected to the Q outputs and decoded to one of sixteen outputs. The output lines can activate an array of sixteen LEDs arranged as a bar graph (for example), or to activate audible alarms or process controls.

The ADC and the DAC open the door to interfacing the computer with practical tasks in the home and in industry. At far less than the cost of available interfaces, you can begin experimenting with the ADC and put your own ideas to use at home. ◇



*"Harold and I are so proud, Mother. Baby encoded his first word today!"*



*A Science Fair Project for Your Youngster:*

# THE ELECTRONIC ELECTROSCOPE

## Indicates when strong electrostatic field exists

BY KEITH KUNDE

PRIOR to 1792 and Allesandro Volta's development of the chemical battery, nearly all electrical experimentation and research involved static electricity. Such static charges are generated on many nonconducting materials through friction with a complementary material, with combinations such as glass and silk, sealing wax and wool, and solidified sulfur rubbed by hand, leading the way in early experiments. Of course, the early experimenters had no means for directly measuring their static charges, but they did observe the forces of attraction and repulsion produced by charged objects. These observations led to the introduction late in the 16th century of the earliest form of electroscope by William Gilbert, who used a pivoting metal pointer to demonstrate the presence of static charges.

Another early form of electroscope used small balls of pith or cork suspended by fine insulating threads so that the forces of attraction and repulsion could be observed through the motions of the charged balls. In 1787, Abraham Bennet invented what became the most familiar form of the device—The Gold-Leaf Electroscope, which consisted of a small brass box having glass windows on two opposing sides, inside of which two strips of very thin gold leaf were suspended face-to-face from a metal rod. The rod passed through a cork in the top of the box and was terminated with a brass disk on its outer end.

A charged object near the disk would cause a similar charge to be