VOL. 2 No. 1 January 1985

COMPUTER DIGEST

NEW KIND OF MAGAZINE FOR ELECTRONICS PROFESSIONALS

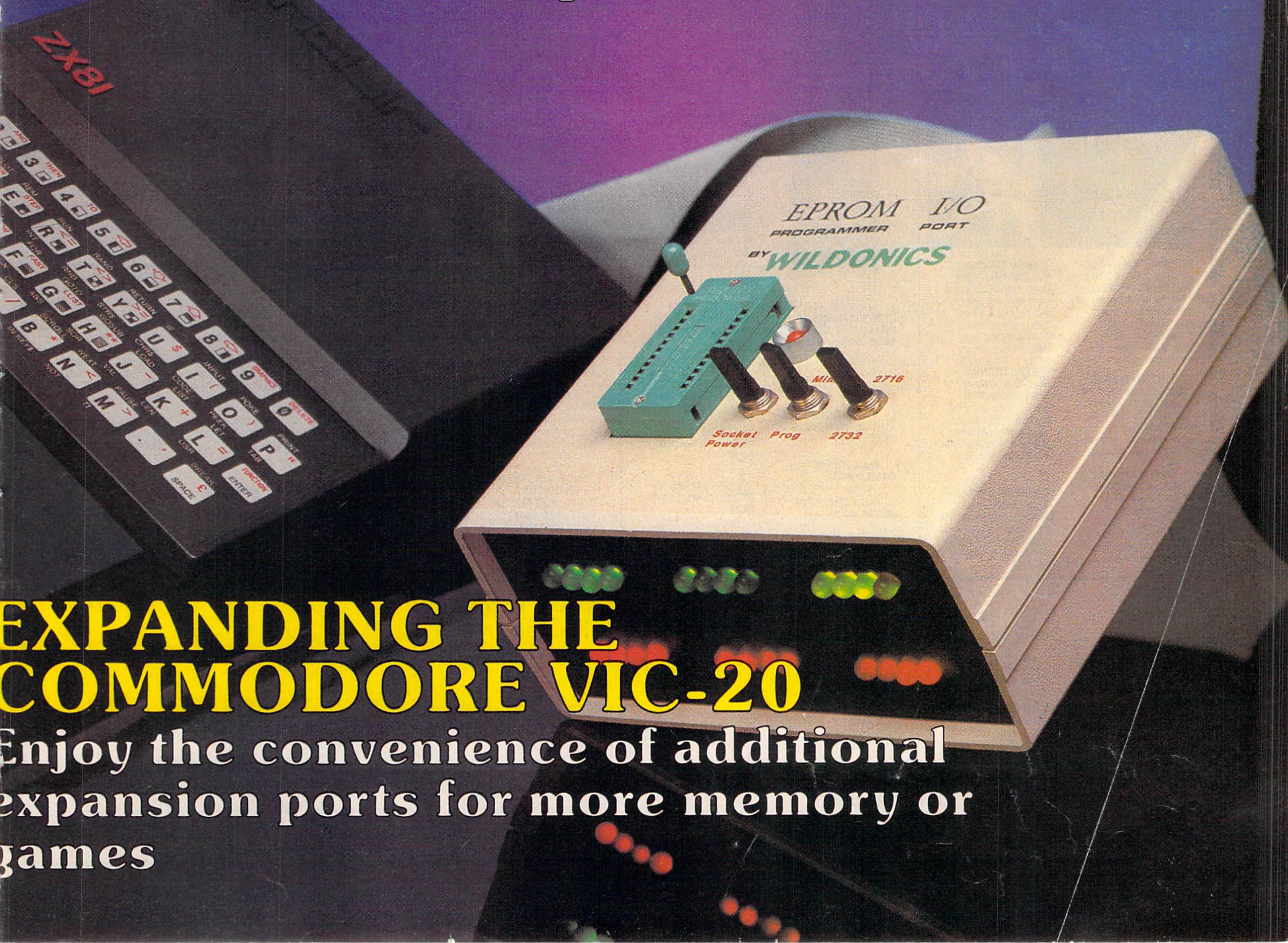
MACHINE CODE DEVELOPMENT SYSTEM

Build this and put your Sinclair ZX-81 to work as a machine code development system, PROM programmer, and PROM emulator

A GERNSBACK PUBLICATION

COMPLETING THE BIOBOX

Build our Biofeedback Monitor and use your computer to keep calm.



EXPANDING THE COMMODORE VIC-20

Enjoy the convenience of additional expansion ports for more memory or games

Electronics Paperback Books

Quality Paperbacks at Affordable Prices

☐ **30 SOLDERLESS BREADBOARD PROJECTS BOOK-1. $5.75.** Whenever possible the same parts are used in several projects. Even a first-time builder can complete these circuits.

☐ **HOW TO GET YOUR ELECTRONIC PROJECTS WORKING. $5.00.** Helps you troubleshoot and repair home-built projects of every description.

☐ **MINI-MATRIX BOARD PROJECTS. $5.00.** A variety of projects that can all be built upon a mini-matrix board that has 10 strips and is 24 holes long.

☐ **MODERN OP-AMP PROJECTS. $5.00.** Wide range of specialized op-amp circuits including lo-noise, lo-distortion, ultra-hi input impedance, etc.

☐ **MULTI-CIRCUIT BOARD PROJECTS. $5.00.** 21 fairly simple projects that can all be built on a single printed-circuit board. All are powered by a 9V battery.

☐ **IC PROJECTS FOR BEGINNERS. $5.00.** Inexpensive digital and linear IC's are used to assemble this selection of circuits intended for the beginner.

☐ **ART OF PROGRAMMING THE 16K ZX81. $6.25.** Topics include full screen, scrolling, PEEK & POKE, plus actual working programs.

☐ **THE 6809 COMPANION. $5.00.** Written for the average assembly language programmer. A discussion of 6809 features & reference work for the 6809 programmer.

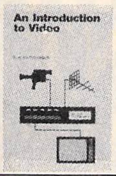
☐ **PRACTICAL COMPUTER EXPERIMENTS. $4.50.** Fills in background to microprocessor by constructing typical computer circuits using discrete logic components.

☐ **ART OF PROGRAMMING THE 1K ZX81. $5.00.** How to use the features of the ZX81 in programs that fit the 1K machine and are still fun to use.

☐ **INTRODUCTION TO BASIC PROGRAMMING TECHNIQUES. $5.00.** Based on author's own experience in learning BASIC and helping others to learn to program.

☐ **A MICROPROCESSOR PRIMER. $4.50.** Painless approach to computing for the beginner. Step-by-step explains computer operations and assembly.

☐ **THE PRE-COMPUTER BOOK. $5.00.** Aimed at the absolute beginner with no knowledge of computers. A non-technical discussion that helps you enter the computer world painlessly.



☐ **AN INTRODUCTION TO VIDEO. $5.00.** Perfect for the person just about to buy a VCR. Discusses pros & cons of the various formats; video discs; videotext, tape copying and more.

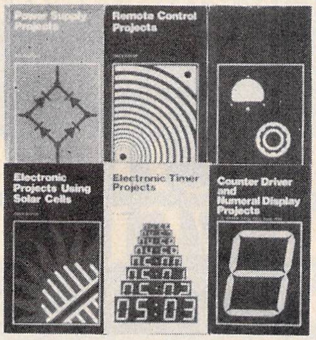
☐ **BASIC & PASCAL. $4.00.** Takes BASIC & Pascal and develops programs in both languages simultaneously.

☐ **FIRST BOOK OF TRANSISTOR EQUIVALENTS & SUBSTITUTES. $3.75.**

☐ **PRACTICAL COMPUTER EXPERIMENTS. $4.50.**

☐ **ELECTRONICS SIMPLIFIED CRYSTAL SET CONSTRUCTION. $4.50.**

☐ **ELECTRONIC HOUSEHOLD PROJECTS. $4.50.**



☐ **POWER SUPPLY PROJECTS. $4.50.** Contains designs and construction details for almost any power supply the experimenter is likely to need.

☐ **REMOTE CONTROL PROJECTS. $5.00.** Radio-control infra-red, visible light, & ultrasonic systems are all included, along with methods of applying them.

☐ **ELECTRONIC TEST EQUIPMENT CONSTRUCTION. $4.50.** Construction details of a wide range of test equipment the experimenter can build at home.

☐ **ELECTRONIC PROJECTS USING SOLAR CELLS. $5.00.** Simple circuits that have numerous applications around the home.

☐ **ELECTRONIC TIMER PROJECTS. $5.00.** Timing circuits for almost any application the experimenter might need. A most valuable reference.

☐ **COUNTER DRIVER & NUMERICAL-DISPLAY PROJECTS. $4.50.** Features applications & projects using various types of numerical-display devices.

☐ **PRACTICAL ELECTRONICS CALCULATIONS AND FORMULAE. $7.50.** A basic reference work that bridges the gap between complicated technical theory & cut and tried methods.

☐ **INTERNATIONAL DIODE EQUIVALENTS GUIDE. $5.75.** Helps you find substitutes for the many different types of semiconductor diodes in use today.

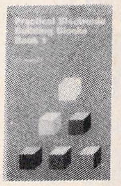
☐ **INTERNATIONAL TRANSISTOR EQUIVALENTS GUIDE. $7.50.** Products of more than 100 manufacturers are listed & cross-referenced with possible replacements.

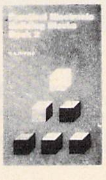
☐ **HOW TO MAKE WALKIE TALKIES. $5.00.** Describes equipment for low-power handheld operation. 112 pages of must reading for the dedicated experimenter.

☐ **ELECTRONIC CALCULATOR USERS HANDBOOK. $3.95.** Presents formulae data, methods of calculation, conversion factors & more from the view of the calculator user.

☐ **LINEAR IC EQUIVALENTS AND PIN CONNECTIONS. $8.25.** Shows equivalents & pin connections of a popular user-oriented selection of linear ICs.

☐ **PRACTICAL ELECTRONIC BUILDING BLOCKS—Book 1. $5.00.** All electronics circuits consist of simple building blocks. When you know how to make the blocks you can easily create your own projects.





☐ **PRACTICAL ELECTRONIC BUILDING BLOCKS—Book 2. $5.00.** Circuits that process signals. Combines with those in Book 1.

☐ **SPECIAL OFFER Book 1 & Book 2 PRACTICAL ELECTRONIC BUILDING BLOCKS. $9.00. Shipping included.** Save $1.75.

☐ **RADIO CONTROL FOR BEGINNERS. $4.50.**

☐ **ELECTRONIC GAMES $4.50.**

☐ **SINGLE IC PROJECTS. $4.25.**

☐ **RADIO CIRCUITS USING IC'S. $4.50.**

☐ **ELECTRONIC HOUSEHOLD PROJECTS. $4.50.**

☐ **ELECTRONIC MUSIC PROJECTS. $4.50.**

☐ **POPULAR ELECTRONIC PROJECTS. $3.75.**

☐ **PROJECTS IN OPTO-ELECTRONICS. $5.00.**

☐ **ANTENNA PROJECTS. $5.00.** Covers practical antenna designs including active, loop & ferrite types that are easy & inexpensive to build.

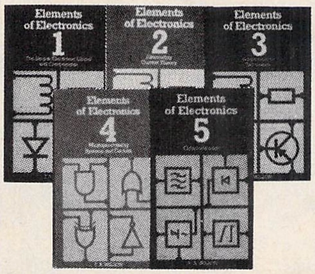
☐ **RADIO STATIONS GUIDE. $4.75.** Comprehensive listing of transmitters around the world. Presents location, frequency, power.

☐ **LONG DISTANCE TV RECEPTION (TVDX) FOR THE ENTHUSIAST. $5.00.** Practical & authoratative introduction to this unusual aspect of electronics.

☐ **CB PROJECTS. $5.00.** A number of useful and interesting designs for CB accessories. Speech processor, interference filter & more.

☐ **CRYSTAL SET CONSTRUCTION. $4.50.** Packed full of easy to duplicate designs for crystal radio receivers.

☐ **AN INTRODUCTION TO RADIO DXING. $5.00.** Listen, in your home, to broadcasts originating thousands of miles away. Tells how you can do it.



☐ **THE SIMPLE ELECTRONIC CIRCUIT & COMPONENTS. $5.75.** All the fundamental theory needed to lead to a full understanding of the simple electronic circuit and its components.

☐ **ALTERNATING CURRENT THEORY. $5.75.** Alternating current theory without which there can be no comprehension of speech, music, radio, or Television.

☐ **MICROPROCESSING SYSTEM & CIRCUITS. $7.50.** A truly comprehensive guide to all of the elements of a microprocessing system.

☐ **SEMICONDUCTOR TECHNOLOGY. $5.75.** Everything you always needed to know about solid-state devices in one volume.

☐ **COMMUNICATIONS. $7.50.** Covers most modern communication systems. Line, microwave, submarine, satellite, digital multiplex, radio & telegraphy.

☐ **FIRST BOOK OF HI-FI SPEAKER ENCLOSURES. $4.50.**

☐ **SOLID STATE NOVELTY CIRCUITS. $3.50.**

☐ **28 TESTED TRANSISTOR PROJECTS. $4.25.**

ELECTRONIC TECHNOLOGY TODAY INC.
P.O. Box 240, Massapequa Park, NY 11762

RE 1/85

Number of books ordered ☐

Total Price of Books .... $________

Sales Tax (NY State Residents) .... ________

Shipping (75¢ 1st two books, 30¢ ea additional) .... ________

TOTAL ENCLOSED .... $________

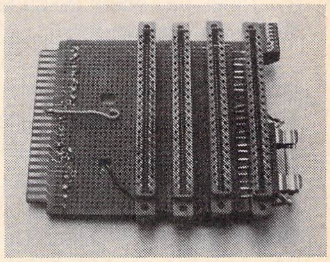
Name________________________________

Address________________________________

City______________________State__________ZIP__________

CONTENTS

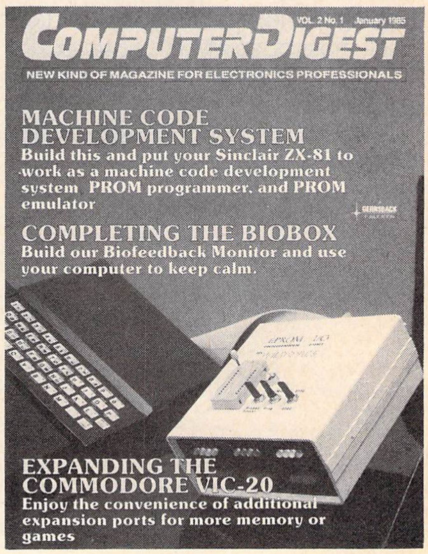
Vol. 2 No. 1 January 1985



Our VIC-20 Expander is shown here ready to go. To find out more about it, **see page 15.**

ON THE COVER

If you own a Sinclair ZX81 or a Timex Sinclair 1000, you already know that programs written in BASIC execute very slowly on those machines. What's more, programming those computers in machine-code can be somewhat less than convenient. This month, we'll show you a project that can make writing and storing machine-code programs a lot easier. What's more, it can double as an EPROM programmer for the ZX81/1000, or an EPROM emulator for another computer system. **See page 7.**



EDITORIAL

Here we go again!

■It's 1985. You know what that means... For the next few weeks, you're still going to be writing "1984" on your checks and letters. A new year takes a little getting used to. But the number of the year isn't the *only* thing that changes.

We're going to be seeing some changes—drastic changes—in our business, too. It seems that there will be drastic new developments announced this year. That computer that you bought because it was the latest, the best, the most up-to-the-minute model, is suddenly going to pale by comparison, and you're going to wonder if you shouldn't trade it in on one of those newer units. Suddenly, the features announced on the new machines will seem essential to you, and you'll wonder how you manage to get along without them.

Don't misunderstand—This is called "built-in obsolesence," and we're in favor of it. It helps keep the economy moving, keeps the money circulating (your money) and the challenge to produce the new and unique in order to compete, keeps the manufacturers on their toes. This results in technological advancement that benefits us all.

Timing is usually a critical factor, too. Remember when the Christmas Season began on December 25th? Now it seems to start on Thanksgiving Day. And the people in Detroit introduce their next year's models during the previous Summer. Happily, computers haven't fallen prey to that gambit as yet. The 1985 models will be coming out now—in 1985. That has both its good and bad aspects: On the positive side of the ledger, those who plan to buy new computers will now be ready to spend their bucks, now that they can shop for the new lines. And others who have been looking for a traded-in "bargain" will therefore find the shelves loaded with choices. And many of us, our Christmas-present money burning a hole in our pockets, will be ready to spend.

People who had been planning to change their jobs waited until after the holidays so they could collect their time off and those Christmas bonuses, but since that's behind us, they *will* be changing jobs now. That's probably going to mean more money for them, a chance to advance for others, and still more openings for those looking for jobs.

Yes, 1985 bodes well for the economy.

And for all of us too.

We, the staff of **ComputerDigest** wish all of our friends a healthy, happy and prosperous 1985

Byron G. Wels

Byron G. Wels
Editor

ComputerDigest is published monthly as an insert in Radio-Electronics magazine by Gernsback Publications, Inc., 200 Park Avenue South, New York, NY 10003. Second-Class Postage Paid at New York, N.Y. and additional mailing offices. Printed in U.S.A.

COMPUTER DIGEST

Gernsback Publications, Inc.
200 Park Ave. South
New York, NY 10003
Chairman of the Board:
M. Harvey Gernsback
President: Larry Steckler

ADVERTISING SALES 212-777-6400
Larry Steckler
Publisher

EAST/SOUTHEAST
Stanley Levitan
Radio-Electronics
200 Park Ave. South
New York, NY 10003
212-777-6400

MIDWEST/Texas/Arkansas/Okla.
Ralph Bergen
Radio-Electronics
540 Frontage Road—Suite 325
Northfield, Illinois 60093
312-446-1444

PACIFIC COAST
Mountain States
Marvin Green
Radio-Electronics
15335 Morrison St., Suite 227,
Sherman Oaks, CA 91403
818-986-2001

LETTERS

DESIGN THOSE AUDIO NETWORKS CORRECTLY!

It was a pleasure to see my article "Computer-Designed Audio Networks" in the November issue of **ComputerDigest**. It was not as pleasant to see that some errors crept into the program listing. The corrected lines follow:

```
5 CLS: PRINT
115 ON N GOTO 120,150: PRINT
    "ERROR - DO OVER": GOTO 110
125 RA=INT(Z*((K-1)/(K+1))):
    RC=INT(Z*((2*K)/
    (K[2-1))):A=Z:B=Z
145 GOSUB 2000: GOSUB 2025:
    GOSUB 2055: GOTO 3000
160 RA=INT((((A+B)*KA)+
    (A-B))/2)
220 GOSUB 1000: GOSUB 1030:
    GOSUB 1050
310 RA=INT((A/SQU(A/B))*
    (((K*SQR(A/B))-1)/K))
530 DB=CINT(ABS(20*(LOG(SQR
    (1/(A/B))/(1+SQR(1-(1/(A/B)))))
    /LOG(10)))+1)
2055 X=60: FOR Y=16 TO 19:
     SET(X,Y): NEXT Y
2060 X=60: FOR Y=24 TO 28:
     SET(X,Y): NEXT Y
```

In line 135, the quotation marks were omitted after the last word, but the program will run without them.—Frank Galdes, Murrysville, PA

Sorry, Frank. Program listings are inherently subject to typesetting/transcription errors—even more so than schematics! If anyone has plans to submit a listing with an article, sending it on an 8-inch SSSD (IBM 3740 format) disk is the best way to avoid any problems.

FLYING ENTHUSIAST

I've become enchanted with the flight simulator on my computer and want to know how valuable this can be toward getting a pilot's license?—Frank Stembo, Dallas, TX.

Frank, as a pilot myself, I can tell you it's very valuable! Student pilots spend a lot of time in the air (and dual instruction costs plenty!) learning the rudiments of instrument flight and navigation. Your flight simulator is saving you a fortune in that way alone. But as one of the old, great aviators once said, "If you want to learn about flying, watch the birds. If you want to learn to fly, get into an airplane!"

COMPUTER VERSATILITY

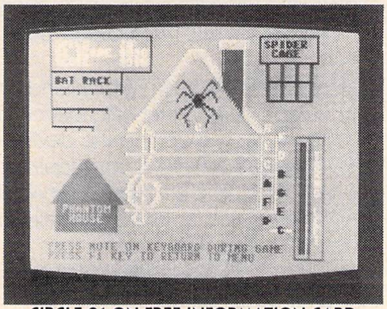
Most people buy computers to solve one particular problem in their lives and only rarely do they look for other applications outside their immediate sphere of interest. Your magazine has made me aware of some of the other things computers can do, and you've broadened my own computer usage. Thanks!—Mort Sabin, Yonkers, NY.

COMPUTER PRODUCTS

For more details use the free information card inside the back cover

EDUCATIONAL SOFTWARE, *The Notable Phantom,* teaches children ages 5–10 basic keyboard (musical) and note-reading skills as they compete against a slew of specters, spiders, and the famous phantom himself.

Haunted-house ghouls lead players through exercises to identify note names and positions on a music staff and keyboard, and to train the ear to identify different tones, depending on which lesson plan the user selects.



CIRCLE 21 ON FREE INFORMATION CARD

Children learn to read music using a songbook of favorite tunes that is included in every game. Budding composers can save their own song creations and play them back later.

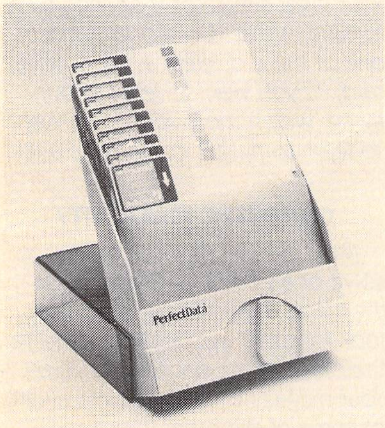
The software comes with a realistic keyboard overlay of black and white notes, more than an octave and a half. The suggested price of *The Notable Phantom* is $49.95.—**Designware,** 185 Berry Street, San Francisco, CA 94107.

PRINTER, the *ThinPrint 80,* is designed for use with portable computers. It is battery-powered, weighs only four pounds, and supports either serial or parallel interfacing to most computers (including Tandy *100* & *PC-2,* Epson *HX-20,* IXO *Telecomputer,* IBM *PC XT, Jr,* and many others).


CIRCLE 22 ON FREE INFORMATION CARD

The *ThinPrint 80* has 80 or 136 columns per line, 40 character-per-second bidirectional printing, 2K buffer memory, and 120 dot-per-inch graphics. It holds 80 pages of 8½" wide paper, produces silent thermal printing, and fits into less than half a briefcase. Its suggested retail price is $279.00, complete with rechargeable batteries, AC adapter, and one roll of paper.—**Axonix Corporation,** 417 Wakara Way, Salt Lake City, UT 84108.

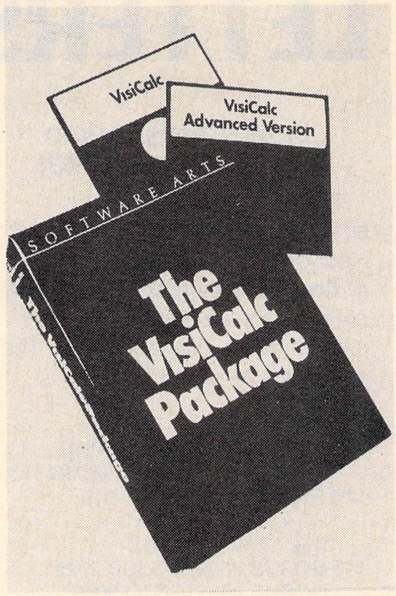
FILING SYSTEM, *Dial 'N File,* is designed for 5¼ diskettes. It is made of high-density, molded plastic and holds up to ten 5¼-inch diskettes and one PerfectData drive-head cleaning disk. When opened, the plastic cover swings into an easel position and becomes a display stand. A clockwise turn of the dial operating the fanning action, places individual diskettes at the user's fingertips, with all diskette labels clearly visible. The *Dial 'N File* case closes with a counterclockwise turn of the dial, and locks securely to protect diskettes from damage and contaminants. Its suggested retail price is $6.95— **PerfectData Corporation,** 9174 Deering Ave., Chatsworth, CA.



CIRCLE 23 ON FREE INFORMATION CARD

SPREADSHEETS, the *VisiCalc Package,* is a two-in-one product containing both a single and a double disk-drive spreadsheet program—*VisiCalc* and *VisiCalc Advanced Version*— for users of the Apple *II* family of personal computers.

The *VisiCalc* disk includes models in home management and finance that can be expanded by the user. They are: checkbook balancing, household budget, individual retirement account analysis, future value of an investment, income averaging, and car-loan payment analysis.

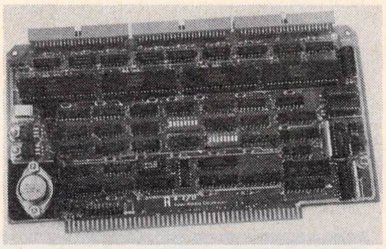


CIRCLE 24 ON FREE INFORMATION CARD

The *VisiCalc Advanced Version* features full word prompts and variable column-width capabilities, as well as date functions, print commands that produce presentation-quality reports, and both 40- and 80-column displays.

The *VisiCalc Package* is priced at $179.00.—**Software Arts Inc,** 27 Mica Lane, Wellesley, MA 02181

COMMUNICATIONS INTERFACE is IEEE-696 (S-100) compatible. It provides a means to connect up to 8 RS232-C devices, regardless of their baud rate, stop bit, and parity configuration. Up to eight of these cards can be used in one system, for a total of 64 channels, with data rates up to 38.4 baud. Also featured are a calendar/clock, switch register, and an encryption device. The calendar/clock is battery backed-up and the device may be disabled in systems requiring several cards. The encryption device is an MMI PAL, and its use is generally for software protection. The interface is priced at $695.00.—**Inner Access Corporation**, PO Box 888, Belmont, CA 94002.


CIRCLE 25 ON FREE INFORMATION CARD

CIRCLE 56 ON FREE INFORMATION CARD

MACHINE CODE DEVELOPMENT SYSTEM

FOR YOUR TIMEX SINCLAIR 1000

Turn your Timex Sinclair 1000 or Sinclair ZX81 into a high-speed, machine-code development system.

MARK W. LATHAM

■By now you may have seen dozens of Timex Sinclair *1000*/Sinclair *ZX81* add-on projects in various electronic magazines. It's not surprising considering that at one time, Timex was shipping 100,000 units a month. While some people are content to fool around with whatever they can hook up to the back of the unit, others have bought real keyboards and extra RAM, hoping to turn their computers into real business or entertainment machines.

If you've ever used a Timex Sinclair *1000* (which we'll simply call a TS *1000* from here on), you know that speed keeps that computer from serving any useful purpose. You could take a short nap while the computer is loading even a 16K program from cassette. Once it's loaded, you run into the other speed problem—execution time. That's because the Z80A CPU spends most of its time updating the video, and, let's face it, the BASIC is too slow, even in the FAST mode. The simplicity of the TS *1000*, which is one of its virtues, is also its downfall.

If you own a TS *1000* and want to turn it into a useful device, why not consider the following: 1) run high-speed machine-language level programs and, 2) store those programs in EPROM.

This project, a machine-code-development-system/EPROM-programmer, will let you do just that. With it, you can use your TS *1000* to load programs from EPROM's, and program EPROM's with data anywhere in the RAM. You will be able to store and recall 4K bytes of battery-backed-up external CMOS RAM. Also, the unit can be disconnected from the TS *1000* and used to emulate an EPROM for a different microprocessor.

You will be able to use the EPROM programmer as a general I/O port, each line of which is monitored by LED's. The LED's are great if you are just learning machine language commands. Of those lines, 20 are available for input/output, while four others are configured as output-only lines capable of sinking 500 mA each. All those lines are available through a socket in the back of the unit and, if you hook them up with a test clip, you will have a five-volt, multi-channel logic monitor with both LED and on-screen viewing. Best of all, the whole EPROM I/O system operates under machine-language level software control, which is, of course, stored in EPROM.

System architecture

The unit is interfaced to the TS *1000* with an 8255 PPI (parallel peripheral interface) I/O port. We could have treated the program socket as a memory space accessed directly by the Z80A, but then we would have had to insert many wait states during the program pulse. Unfortunately, there is no way the CPU can refresh dynamic RAM during waits so that option is out. What we must do then is create a second bus system as shown in Fig. 1, the schematic diagram.

Gates IC1-c and IC2-c allow the Z80A to access the 8255 when A7 and $\overline{\text{IORQ}}$ are low. (A7 is included to ensure that there will be no erroneous writes to the 8255.) If we leave the 8255's AØ and A1 lines set for all I/O operations, the computer's monitor system won't crash during I/O operations regardless of whether the computer is in the fast or slow mode. The A4 and A5 lines of the Z80A are used to control the 8255's AØ and A1 inputs, so, in hexcidecimal, the I/O addresses will be Ø3H, 23H, and 33H.

The 8255 has three eight-bit ports, one of which is bit-addressable. Port C (PB4–PB7) will function as the secondary bus control outputs. Port B (PBØ–PB7) will function as the data I/O port, and ports C (PC3–PCØ) and A (PAØ–7) will function as address outputs Ø–11, respectively. (The reason PC3–PCØ are used in reverse as AØ–A3 is twofold; that both simplifies circuit board layout and arranges the bus and LED's for use as a logic monitor, as you will see later.)

When the 8255 is reset (either by the computer or on power up) all the ports are configured as inputs. Any time those ports are changed from inputs to outputs, or vice-versa, all the port registers are reset. That presents a problem for the control lines in our secondary bus system because those lines must remain high (set) until a memory access is desired. Transistors Q1–Q4 are used to alleviate that problem. If a port's input or output is low, the corresponding transistor output is high, holding the control line secure. If the data in the CMOS RAM is of no importance, then those transistors may be used as high current outputs, capable of sinking up to 500 mA each.

The CMOS RAM, IC9 and IC10, and the CMOS one-of-eight decoder, IC7, provide 4K of data storage for program saving and ROM emulation. The decoder

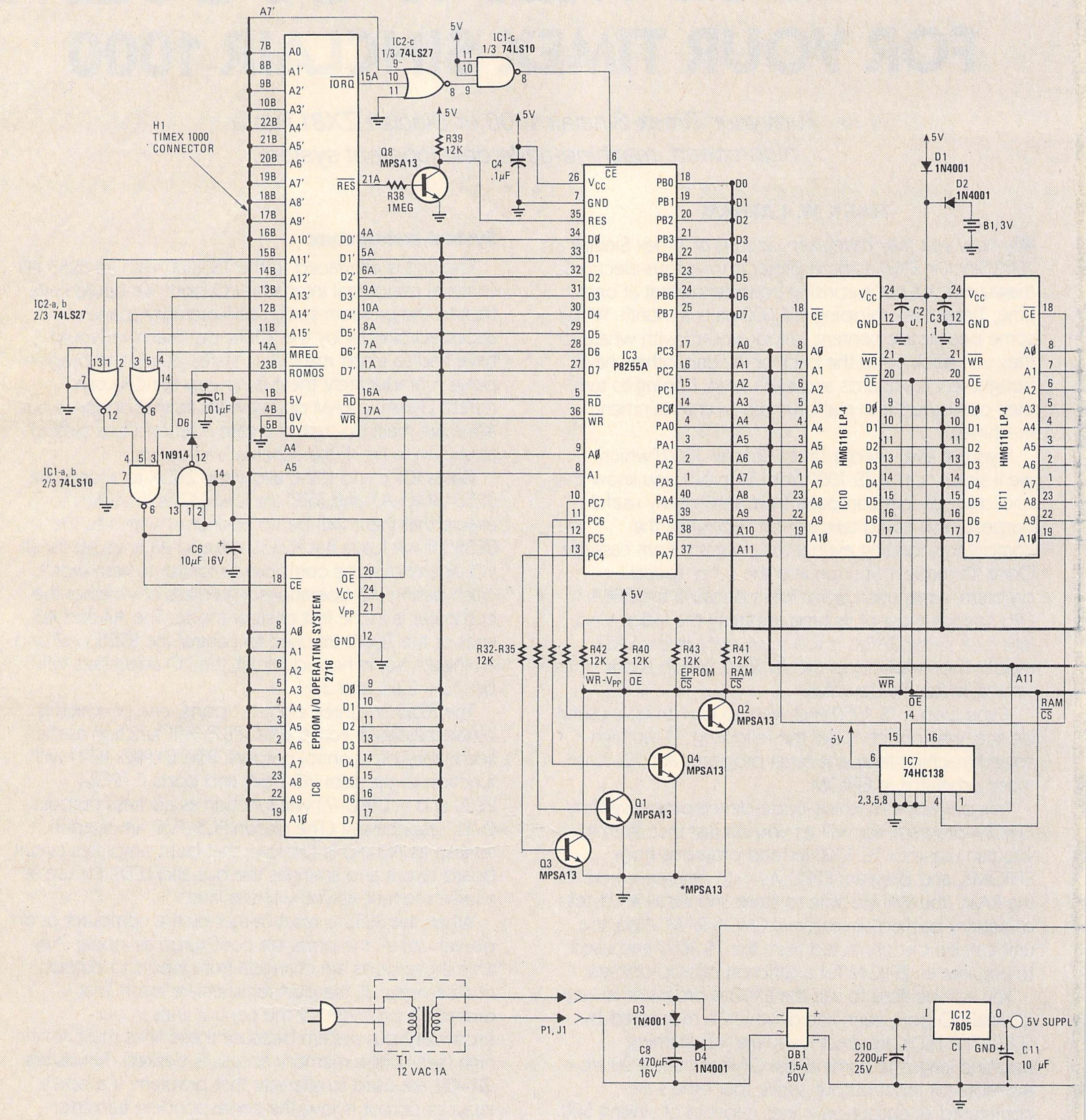
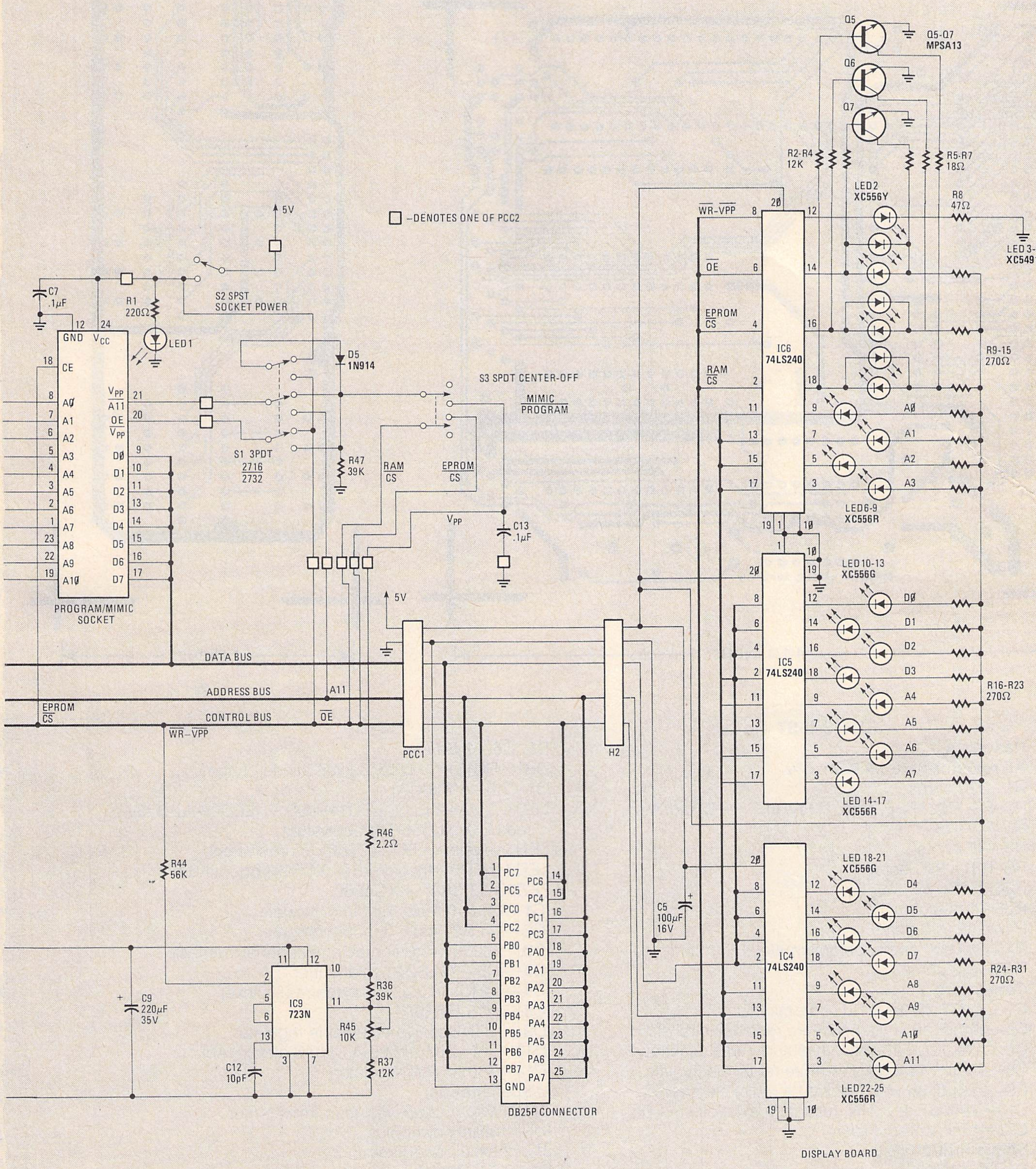
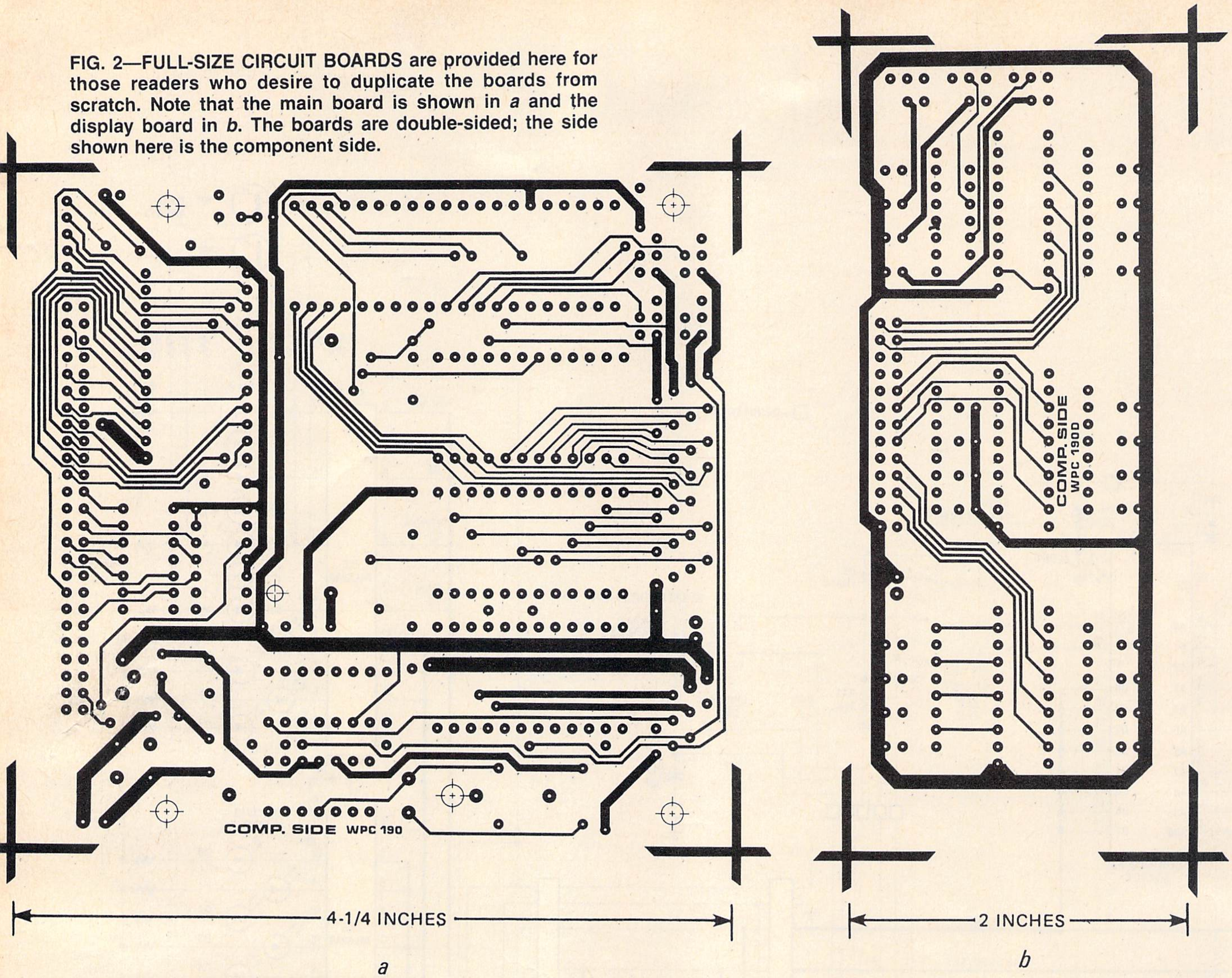


FIG. 1—COMPLETE SCHEMATIC DIAGRAM. Reference the diagram carefully while reading the text, as it helps clarify some of the more-complicated points.



Q5-Q7
MPSA13
Q5
Q6
Q7
R2-R4
12K
R5-R7
18Ω
LED2
XC556Y
R8
47Ω
LED 3-5
XC5491
5V
–DENOTES ONE OF PCC2
C7
.1µF
R1
220Ω
LED1
S2 SPST
SOCKET POWER
D5
1N914
S3 SPDT CENTER-OFF
MIMIC
PROGRAM
S1 3PDT
2716
2732
R47
39K
RAM
CS
EPROM
CS
VPP
C13
.1µF
PROGRAM/MIMIC
SOCKET
GND
VCC
CE
WR-VPP
OE
EPROM
CS
IC6
74LS240
R9-15
270Ω
LED6-9
XC556R
LED 10-13
XC556G
IC5
74LS240
R16-R23
270Ω
LED 14-17
XC556R
DATA BUS
ADDRESS BUS
CONTROL BUS
A11
PCC1
H2
R46
2.2Ω
R44
56K
C5
100µF
16V
LED 18-21
XC556G
IC4
74LS240
R24-R31
270Ω
LED22-25
XC556R
C9
220µF
35V
IC9
723N
R36
39K
R45
10K
R37
12K
C12
10pF
DB25P CONNECTOR
DISPLAY BOARD

FIG. 2—FULL-SIZE CIRCUIT BOARDS are provided here for those readers who desire to duplicate the boards from scratch. Note that the main board is shown in *a* and the display board in *b*. The boards are double-sided; the side shown here is the component side.



PARTS LIST

Resistors

All resistors are 1/4 watt, 5%

R1—220 ohms
R2–R4, R32–R35, R37, R39–R43—12,000 ohms
R5–R7—18 ohms
R8—47 ohms
R9–R31—270 ohms
R36, R47—39,000
R38—1 megohm
R44—56,000 ohms
R45—10,000 ohms, potentiometer, PC mount
R46—2.2 ohms

Capacitors

C1–C4, C7, C13—0.1µF, ceramic disc
C5—100µF, 16 volts, miniature radial electrolytic
C6, C11—10µF, 16 volts, miniature radial electrolytic
C8—470µF, 16 volts, miniature radial electrolytic
C9—220µF, 35 volts, miniature radial electrolytic
C10—2200µF, 25 volts, miniature axial electrolytic
C12—10pF, ceramic disc

Semiconductors

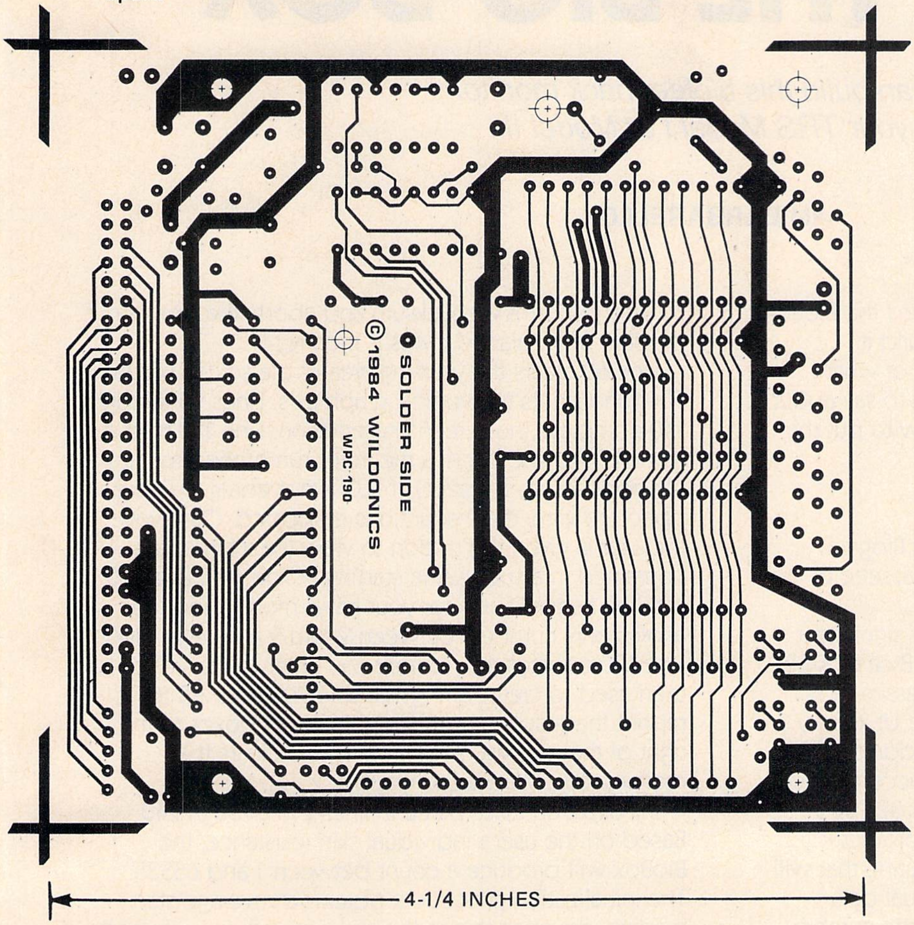
D1–D4—1N4001
D5, D6—1N914
DB1—RB151 1.5-amp, 50 volt, diode bridge
Q1–Q18—MPSA13
LED1, LED6–LED9, LED14–LED17, LED22–LED25—red LED, XC556R or equivalent
LED 2—yellow LED, XC556Y or equivalent
LED3–LED5—tricolor LED, XC5491 or equivalent
LED10–13, 18-21—XC556G
IC1—74LS10 triple 3-input NAND gate
IC2—74LS27 triple 3-input NOR gate
IC3—P8255 programmable peripherial interface
IC4—IC6—74LS240 octal buffer
IC7—74HC138 3 to 8 decoder/multiplexer
IC8—2716 EPROM
IC9—723N positive adjustable regulator
IC10, IC11—HM6116LP-4 CMOS static RAM
IC12—7805 5-volt regulator

Miscellaneous

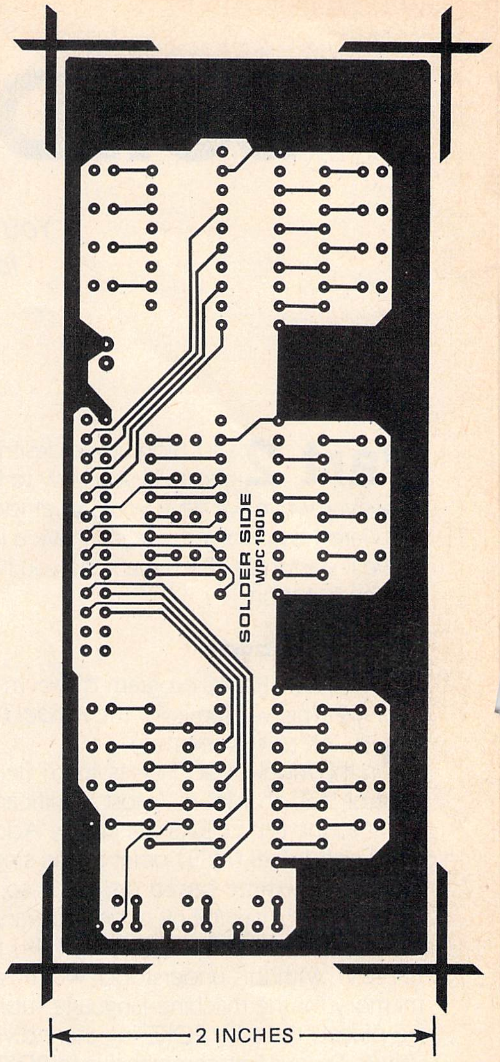
T1—12VAC, 1-amp, wall-plug transformer
P1—coaxial power plug
J1—coaxial power jack
S1—3PDT switch

reads $\overline{RAMCS}$ (PC7) and A11 to select the appropriate memory IC. Those three IC's are powered by either the five-volt supply through D1 or the lithium three-volt battery through D2. Pin 6 of the decoder monitors the five-volt supply and disables the RAM when the power is off.

FIG. 3—THE SOLDER SIDE OF BOTH BOARDS (the main board is shown in *a*; the display board in *b*) is given here, also full size. Both boards can be etched at once and then cut apart.



a



b

S2—SPST switch
S3—DPDT switch, center-off
H1—50-contact, right-angle header
H2—26-contact header
PROGRAM SOCKET—24-pin ZIF socket with extender pins (or wire wrap socket)
PC boards, IC sockets, enclosure, hardware, ribbon cable, card-edge connector, DB-25 connector, etc.

The following are available from Wildonics Computer Technologies, P.O. Box 1763, Boise, ID, 83701: Complete kit of all parts including power supply, all connectors, lithium battery, PC boards, and case (does NOT include 2716 EPROM with Operating System), $149.95; 2716 EPROM with Operating System, $19.95; set of drilled and etched PC boards only $19.95; Assembled and tested unit with Operating System Software, $219.95. Shipping, handling and insurance, $3.00 for EPROM with software or PC boards only. $6.00 for complete kit or assembled unit.

With S3 set for MIMIC and the 8255's ports all configured as inputs, a secondary CPU can directly access the CMOS RAM through the PROGRAM socket. Setting S3 for MIMIC simply OR-ties the $\overline{RAMCS}$ and the $\overline{EPROMCS}$ lines and bypasses V_{PP}-blocking diode D5. Resistors R38 and R39, and transistor Q8, which normally act as an inverter for the $\overline{RESET}$ signal, hold the 8255 reset if the EPROM-I/O unit is used apart from the ZX81 during a mimic operation.

When S3 is set to PROGRAM, the output of the V_{PP} switching regulator, IC9, is connected to the appropriate EPROM I/O pin. $\overline{WR}$ (PC7) controls the regulator's output by sourcing the base of the regulator's current limiting transistor. For that application, that transistor's emmitter is connected to ground. Capacitor C12 is connected to the frequency-response pin to slow the V_{PP} rise and fall times. Diodes D3 and D4 and capacitors C8 and C9 act as a voltage doubler to provide 30 volts at 60 mA to the regulator's input.

All the bus lines can be monitored with the display board. Three 74LS240's, IC4–IC6, power the LED's. Red LED's (LED6–LED9, LED14–LED17, and LED22–LED25) are used for the the address lines and the LED's for the data lines (LED10–LED13 and LED18–LED17) are green. Those LED's will light when the corresponding bus lines are high or high-impedance. The yellow LED (LED2) will light if the $\overline{WR\text{-}V_{PP}}$ line is low.

While we are out of space, we're not our of things to say. We'll finish up next month.

BUILD THE BIO-BOX

You can build this biofeedback monitor for your TRS Model I or Model III.

JIM BARBARELLO

Part 2 Last month, we described the BioBox and told you how to build it. However, while we gave you a brief idea of what software was involved, we still have a lot to say about it. We'll start there. Then we'll tell you how to put the BioBox in action.

The BASIC program

The BioBox BASIC program is shown in Program Listing 2. This version is for the *Model I*, cassette or disk based, 16K to 48K memory.

On the *Model I*, addresses 16561 (least significant byte or LSB) and 16562 (Most Significant Byte or MSB) point to the top of BASIC memory. Addresses 16527 (LSB) and 16528 (MSB) point to the single USR entry point in a cassette-based system. Also, address 16561 is always 255, but 16562's contents vary according to the available memory (127 for 16K, 191 for 32K and 255 for 48K). With this understood, we must protect memory for the machine-language subroutine that will be placed there by POKE-ing the individual data values. So our first command is to POKE the number 215 into location 16561. This reserves an ample 40 bytes for our subroutine. Next, we clear 1000 to reserve string storage space and reset BASIC pointers. Line 10 also defines an error handling routine starting at line 550.

Line 20 is valid for a disk system. J is set to two bytes past the protected memory start. The second statement adjusts J if it is greater than 32767, so it can be used in the POKE statements. Finally, we jump over line 30 (which is used only for cassette-based systems) and continue execution at line 40. If we run this program on a cassette-based system, the DEFUSR statement in line 20 will cause an error, branching execution to line 550.

Line 560 checks to see if the error has occurred in line 20 (indicating that this is not a disk system). If so, we resume execution at line 30. In this manner, we can have the program decide which line to use, based on the system configuration.

The first statement in line 30 is for older *Model I's*. Those units had a software error in the ROM which affected the DATA pointer, making DATA reads impossible. POKE-ing 255 into location 16553 corrects this. Line 30 then sets the USR entry point and calculates the starting location to begin POKE-ing the machine language code. Line 40 prints a heading, while line 50 POKES the subroutine code into memory. Line 50 also performs a checksum and aborts the program if the sum of all placed bytes is incorrect.

Line 60 clears the working area of the screen (GOSUB 900) and prints the vertical graph axis. Lines 90 through 140 complete the screen presentation. Line 150 tests a flag (FLG) to see if this is the initial run of the program (as opposed to a restart). If FLG is not equal to 0, the option to view the instructions is skipped. Otherwise, the user is given the option to view the instructions contained in a subroutine starting at line 600. Line 170 sets the flag and asks for your initial mood. Line 180 allows only numbers between 2 and 9 as a valid input. Line 190 equates the number you enter to CC, which is then used to create string AB. Line 200 uses AB to reprint the graph presentation minus all boxes to the right of the row you specified. It also clears the message line (above the graph).

Initialization takes place in lines 210 through 240. Based on the user's individual skin resistance, the BioBox will produce a count between 1 and 65535. The initialization procedure obtains an average start reading, equating this to the initial mood you selected. It then sets upper (maximum calm) and lower (maximum tenseness) count limits and a change increment (INC). INC is the maximum change in count that will cause one box to be removed or added. The USR call in line 210 is not included in the average count, but simply insures that the BioBox is reset before sampling begins. Line 240 erases the INITIALIZING message before proceeding to the actual biofeedback monitoring of lines 250 through 410.

First we print a box at the present position, and a period (dot) directly above it. Then line 260 samples the BioBox and, if 0 is returned, creates an error to pass execution to the error-trap routine at line 550. Line 270 increments the time (XT) and line 280 polls the keyboard to see if a Restart or End was requested (GOSUB 740). If not, the current square is blanked out (this creates the blinking effect). Line 300 checks to see if the count change is less than one increment. If so, execution branches to line 410 where a delay proportional to the current count is created before returning to line 250 for the next sample.

If the change is greater than one increment, line 310 checks to see if the count is increasing (less tense) or decreasing. For an increasing count, line 330 increases XO by one increment. Then R and C are checked and adjusted if necessary to point to the top of the preceding column (if the last square in the current

PROGRAM LISTING 2

```
10 POKE 16561,215:CLEAR 1ØØØ:DEFSTR A: DEFINT
   C,I,R:DIM T(5Ø,3) ON ERROR GOTO 55Ø
2Ø J=217+PEEK(16562)*256:J=J+(J>32737)
   *65535:DEFUSR=J+1:GOTO 4Ø
3Ø POKE 16553,255:POKE 16526,217:POKE 16527,PEEK
   (16562): J=216+PEEK(16527)*256:J=J+(J>32737)
   *65535
4Ø CLS:PRINTTAB(14);"B I O F E E D B A C K
   M O N I T O R" :PRINTSTRING$(63,131):
   PRINTTAB(18);"(c) 1983 by J. J. BARBARELLO"
5Ø FORI=1TO32:READ N:POKE J+I,N:K=K+N:NEXT:IF
   K<>3647 THEN PRINT @536, "CHECKSUM
   ERROR.":END
6Ø GOSUB 9ØØ;FORI=1TO1Ø:PRINT" ";CHR$(157)
   :NEXT:PRINT" ";CHR$(141)
7Ø DATA 243,62,1,211,255,6,64,16,254,62,Ø,211,255,17,1,
   Ø,33
8Ø DATA Ø,Ø,219,255,254,255,4Ø,3,25,48,247,251,
   195,154,1Ø
9Ø PRINT@839,CHR$(14Ø);:FORI=1TO1Ø:PRINTSTRING
   $(2,14Ø):CHR$(142);STRING$(2,14Ø);:NEXT
1ØØ AL=" ":A=STRING$(3, 143)+" ":FORI=1TO1Ø:
    AL=AL+A:NEXT
11Ø FORI=1TO1Ø:PRINT@2Ø1+(I-1)*64,AL;:NEXT
12Ø PRINT@9Ø5, ;:FORI=1TO1Ø:PRINTUSING"##
    ";I;:NEXT
13Ø PRINT@968, "CALM";TAB(52);"TENSE";
14Ø T$="^TENSE":FORI=1TO6:PRINT@259+I*64,
    MID$(T$,I,1);:NEXT
15Ø IF FLG>Ø THEN 17Ø
16Ø PRINT@98Ø, "INSTRUCTIONS? (Y/N)...";:GOSUB 8Ø
    Ø:IF AI="N" THEN PRINT @98Ø,STRING$(25,32);
    ELSE GOSUB 6ØØ
17Ø FLG=2:PRINT@145, "SELECT INITIAL MOOD
    (2-9)...";
18Ø AR=INKEY$:IF AR=""THEN18Ø ELSE GOSUB 76Ø:IF
    VAL(AR) <2 OR VAL (AR)>9 THEN 18Ø
19Ø PRINTAR;:CC=VAL (AR) :R=2ØØ:C=CC*5:
    AB=STRING$((1Ø-CC)*5,32)
2ØØ FORI=Ø TO9:PRINT@R+C+64*I,AB;:NEXT:
    PRINT@145,STRING$(5Ø,32); :C=C-5)
21Ø PRINT@985, "INITIALIZING...";:Y=Ø:FLG=2:XT=Ø:
    X=USR(Ø)
22Ø FORI=1TO5:PRINT@R-62+C, ".";:X=USR(Ø):
    PRINT@R-62+C," ";:IFX<Ø THEN X=65534+X
23Ø Y=Y+X:NEXT:XO=Y/5:INC=XO/(1Ø*(CC+5))
    :XL=XO-CC*INC*1Ø
24Ø PRINT@985,STRING$(15,32);
25Ø PRINT@R+C+1,A;:PRINT@R-62+C, ".";
26Ø X=USR(Ø):IF X<Ø THEN X=X+65536 ELSE IF X=Ø
    THEN ERROR 1
27Ø XT=XT+X/3ØØØØ
28Ø GOSUB 74Ø
29Ø PRINT@R-62+C, " ";
3ØØ IF ABS (XO-X) <INC THEN 41Ø
31Ø IF X<XO THEN 37Ø
32Ø IF X<XO THEN 4ØØ
33Ø PRINT@R+C+1,STRING$(5Ø-C,32);:XO=XO+
    INC:R=R+64:IF R=84Ø THEN R=2ØØ:C=C-5
34Ø IF C>-1 THEN GOSUB 5ØØ:GOTO 32Ø
35Ø PRINT@468, "MAXIMUM CALM ATTAINED.";:
    PRINT@525, "PRESS <R> TO RESTART, OR <E>
    TO END...";
36Ø GOSUB 74Ø:GOTO36Ø
37Ø R=R-64:IF R=<136 THEN R=776:C=C+5
38Ø IF C>45 THEN C=45:R=2ØØ:GOTO 4ØØ
39Ø XO=XO-INC:PRINT@R+C+1,A;:GOSUB
    5ØØ:GOTO 31Ø
4ØØ X=XO
41Ø FORI=1TO(X-XL)*25Ø/XL:NEXT:XT=XT+I/5ØØ:GOTO
    25Ø
51Ø FOR Z=1 TO 5Ø:NEXT:XT=XT+.25:RETURN
53Ø PRINT@98Ø, "PRESS ANY KEY TO CONTINUE";
54Ø AI=INKEY$:IF AI="" THEN 54Ø ELSE RETURN
56Ø IF ERR=56 THEN PRINT@985, "PRINTER
    ERROR";:STOP ELSE IF ERL=2Ø THEN RESUME 3Ø
57Ø PRINT@966, "ERROR OCCURRED. PRESS <R> TO
    RESTART, <E> TO END ..";
58Ø AR=INKEY$:IF AR=""THEN58Ø ELSE
    NU=ASC(AR):IF NU>91 THEN AR=CHR$(NU-32)
59Ø IF AR<> "R" AND AR<>"E" THEN 58Ø ELSE
    PRINT@966,STRING$(55,32);:RESUME 76Ø
61Ø GOSUB 89Ø:PRINT@26Ø, "The Biofeedback System I
    measures and displays your changes in mood. Before
    beginning, check that the hardware interface is attached,
    and power is applied."
62Ø PRINT" Next, attach one BioProbe to your index finger
    above the first joint. Then place the remaining BioProbe
    on your middle finger above its first joint."
63Ø PRINT" When you have finished reading these
    instructions, you'll be asked the question ";CHR$(34);"
    SELECT INITIAL MOOD (2-9)...";CHR$(34);". Select a
    number between 2 (CALM) and 9 (TENSE). ";
64Ø PRINT"If you're in an average mood, select 5. If you're
    calmer, try a lower number (like 3). Otherwise, select a
    higher number (like 8).";
65Ø GOSUB 53Ø:GOSUB 89Ø
66Ø PRINT@26Ø, "Your mood is represented by the 1ØØ
    blocks. When you select your initial mood, the higher
    tension-indicating blocks will disappear.";
67Ø PRINT"The object is to relax and in the process make
    all the blocks disappear. If you increase tension the
    blocks will begin reappearing. A blinking dot will remind
    you where you currently are.";
68Ø PRINT"The more tense you get, the faster it blinks. The
    calmer you get, the slower it blinks."
69Ø PRINT"If a fault occurs in the BioBox (EX: BioProbes
    come loose, power not applied), a message will appear
    and allow you to re-start by pressing <R>. If you wish
    to restart at any other time, press <R>.";
7ØØ PRINT"When you wish to end the session, press <E>."
71Ø GOSUB 53Ø:FLG=2:GOTO 6Ø
75Ø AR=INKEY$:IF AR=""THEN RETURN
76Ø IF AR="R" OR AR="r" THEN PRINT@128, TAB(24);
    "R E S T A R T";TAB(6Ø);:GOSUB 84Ø:GOTO 6Ø
77Ø IF AR="E" OR AR="e" THEN GOSUB 84Ø:GOTO 99Ø
    ELSE RETURN
8ØØ AI=INKEY$:IFAI=""THEN8ØØ ELSE NU=ASC(AI)
81Ø IFNU>91THEN NU=NU-32
82Ø AI=CHR$(NU):IFAI<> "Y"ANDAI <> "N"THEN8Ø
    ØELSERETURN
84Ø REM** STORE RESULTS
85Ø PRINT@98Ø, "STORE RESULTS? (Y/N)...";:GOSUB
    8ØØ:PRINTAI;
86Ø IF AI="Y" THEN
    S=S+1:T(S,1)=CC:T(S,2)=XO-(Y/5))/
    INC:T(S,3)=XT
87Ø RETURN
9ØØ PRINT@128," ":FORI=1TO11:PRINT" ":NEXT:
    PRINT@96Ø STRING$(63,32);:PRINT@192,;:
    RETURN
1ØØØ ST=1:TN=Ø:GOSUB 9ØØØ:IFS=Ø THEN 1Ø5Ø ELSE
     ON ERROR GOTO 1Ø5Ø
1Ø1Ø PRINT@64, TAB(27);"R E S U L T S"TAB(63):
     PRINT@129, "TRIAL #" TAB(2Ø)
     "START"TAB(32) "END"TAB(44) "TIME"TAB(56)
     "FACTOR":PRINTSTRING$(62,"-")
1Ø2Ø FORI=ST TO S:PRINTUSING" ##";I;:PRINTTAB(21);
     :PRINTUSING"###";T(I,1)*1Ø;:PRINTTAB(32);
     :PRINTUSING";T(I,1)*1Ø-TI,2);:PRINTTAB(43);
     :PRINTUSING"####";T(I,3);
1Ø3Ø PRINTTAB (55);:PRINTUSING"####.##";T(I,3)/
     T(I,2)
1Ø4Ø TN=TN+1:IF TN=1Ø THEN GOSUB 53Ø:
     TN=Ø:ST=ST+1Ø:GOSUB 9ØØ:GOTO 1Ø1Ø
     ELSE NEXT
1Ø5Ø PRINT:PRINT"RESTART? (Y/N)...";:GOSUB 8ØØ:IF
     AI="Y" THEN PRINT@64,STRING$(63,131);
     STRING$(65,32):ON ERROR GOTO 55Ø:GOTO 6Ø
1Ø6Ø END
```

column is being removed). If C has not been decremented past 0, we jump to the subroutine at line 500, where a fixed delay is created and the time is updated. Then we return to line 320. This procedure continues until the difference between X and XO is less than one increment. The same procedure is followed in lines 370 through 390 for a decreasing count.

If at any time, all squares are removed, execution passes to line 350 where the MAXIMUM CALM ATTAINED message is displayed, and we are allowed to (R)estart or (E)nd. Lines 500 through 710 contain various subroutines, including that to display the instructions. The Restart/End subroutine begins at line 740. This subroutine is used throughout the program and allows one to restart or end at almost any time. It also calls another subroutine that gives you the option to save the results of any trial for later presentation. (STORE RESULTS, beginning at line 840).

The END routine begins at line 990. Line 1000 passes execution to Line 1050 (Restart?) if no data are present, or branches to the error trap if any error occurs. Otherwise, line 1010 proceeds to display the results previously stored in the T array. Notice that the "Factor" is a relative measure of results, since it reflects number of squares removed per unit time. Since up to 50 trials can be stored, the FOR/NEXT loop starting at line 1020 prints results in groups of 10 maximum, waits for you to press any key, and then continues. Line 1050 allows you to restart or truly end. In this manner, you can select the END function at any time, review your results and then RESTART to continue monitoring.

Using the biobox

Select a quiet, comfortable area (around 70 degrees F). Relax by loosening tight clothing, removing your shoes, etc. Sit in a comfortable position that provides arm and elbow support. Make sure your hands are clean and dry.

Type in, save and then RUN the BIO program. After the initial screen has been displayed, place the black cassette cable plug in J2 (out) and the large grey cassette cable plug in J1 (in). The small grey plug is not used. Place S1 (power) in the ON position.

The display consists of a title at the top, an underline, an "option/status" line, the biofeedback graph and a command line. At this point, the status line contains a copyright notice and the command line is asking "Instructions? (Y/N)..." Press "Y." The screen will clear below the title and the first page of instructions will be displayed. When done reading, press any key to continue (as instructed at the bottom of the screen) to read the second page of instructions. When you press any key again, you are returned to the opening screen. But note that the copyright notice is replaced by the question "SELECT INITIAL MOOD (2-9)..." You would have been brought to this point immediately if you responded N (no) to the "INSTRUCTIONS" question.

Now place one bioprobe on your index finger, and the other bioprobe on the middle finger of the same hand. The bioprobe foil should contact the fingerprint. Set the BioBox's ON/OFF switch to the ON position, and press "5." Columns 6 through 10 will disappear. The message "INITIALIZING" will appear at the bottom, and a dot (period) will appear over the top box on the last row (5th row in this example). If the BioBox is not working properly, (power not on, bioprobe not attached, skin resistance too high, etc.) the message "ERROR OCCURRED. PRESS (R) TO RESTART, (E) TO END..." will appear at the bottom. Correct the problem (power up the BioBox, attach probes, clean fingers, etc.) and press (R) to try again. You will be asked if you want to "STORE RESULTS? Y/N..." If you have completed a valid session, you would select "Y." If you encountered an error (or simply do not want to store results) press "N." The message "R E S T A R T" will appear at the top of the screen and the original display will be provided.

Select an initial mood between 2 and 9. The dot will blink five times, and the "INITIALIZING" message will then disappear. You are now in the biofeedback monitoring mode. Make a fist; boxes will begin to be added. Release the fist; boxes will disappear (in an actual session, you should keep your hand stationary). The object is to remove all boxes. If you do, the message "MAXIMUM CALM ATTAINED. PRESS (R) TO RESTART, OR (E) TO END..." will appear in the middle of the screen. Whichever you choose, the message "STORE RESULTS? (Y/N)..." will appear at the screen bottom. Note that during monitoring you may press (R) to restart or (E) to end at any time, but you may have to hold the key down for a second or so before it is recognized. When you select (E) you will see the RESULTS screen. The RESULTS display contains five columns, labelled TRIAL #, START, END, TIME and FACTOR. For each trial, the START and END columns show the number of squares you started and ended with. For instance, if you selected "6" as your initial mood and acheived maximum calm, the START indication would be 60 (6 columns × 10 squares/column = 60) and the END indication would be 0. The next column indicates the elapsed time of the session (not seconds, but relative units of time). The final column gives an indication of how well you did. It is a ratio of the number of squares removed per one unit of time. The object is to get this number as close to zero without going negative (which indicates squares were added, not removed.)

If there are more than 10 results stored, they will be shown in pages of 10. When all results have been displayed, you will be given the option to "RESTART? (Y/N)..." By pressing "Y" you can continue monitoring. (This allows you to periodically check your progress and then return to monitoring.) If you select "N," the program will end. As currently written, the data is not permanently saved. Depending on your individual system and requirements, a short subroutine may be added to save the data to a tape or disk file.

Summing it up

The BioBox can turn your Model I or III into a computerized biofeedback monitoring system, and may even help you to reduce everyday stress and tensions. But don't limit it strictly to biofeedback monitoring. Try it as a lie detector at your next party. Just make sure you don't become the subject!

VIC-20 EXPANDER

Build this expansion port for your VIC-20.

JIM STEELE

■If you own a Commodore *VIC-20*, you're probably tired of switching memory-expansion modules and game cartriges in and out of the user port. You might have considered buying one of those port expanders you've seen advertised. They are certainly a possible solution—you can switch between several cartridges at the flip of a switch—but they're expensive. We'll show you a less-expensive alternative—building your own port expander.

Additional ports

While the expander module shown here will provide three additional ports with another available for future expansion, there is no reason why this selfsame system could not be further expanded upon almost to an infinite number of ports, limited only by your own requirements and your own pocketbook. There are actually two ways to go.

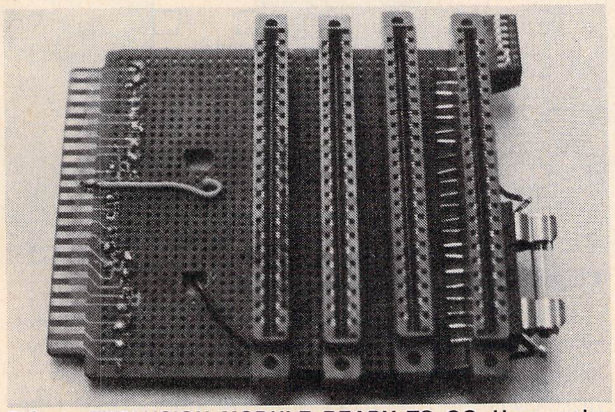


FIG. 1—EXPANSION MODULE READY TO GO. Here, we're looking down at the top of the board.

One way is to make additional expanders, as shown here, and simply plug the second expander unit into the open port on the first one. However, if you anticipate the need for several more ports than would be furnished by this unit, you can readily "expand the expander" by adding additional ports wired in the same configuration as these are.

The result will be even more versatility.

The justification for this expander is simply in its added conveniece to the user. Before the expander, it was necessary (within limiations) to pull a cartridge and replace it with another when cartridges needed changing. If you rarely if ever change cartridges, the expander will seem a mere nicety that you could probably do as well without. However, if you're constantly changing cartridges, as would be the case when you're using your computer predominatly for game-playing, the expander becomes a vital and important tool, as you leave *all* the cartridges plugged in, and flip switches to change from one to another. It makes life a great deal simpler.

You can put the expander together for about $20.00—perhaps less if you have some of the parts around. But it will work just as well as the commercial models that cost up to five times as much. The expander we'll show you was built to accept three cartridges, with a fourth available for future additions. It is fully switchable and it's fused to protect both the VIC and cartridges.

Easy to build

Even if you don't have much experience building electronics projects, you shouldn't have too much trouble with the expander. The hardest task is the point-to-point wiring, but you can get around that by designing a printed-circuit board. Whatever method you use, you should be able to finish everything up in a weekend.

To begin with, you will need a general-purpose plug board with a 22/44 edge connector. Such boards are available from many sources, including Radio Shack. Next, you will need three or four wire-wrap 44-pin card-edge connectors. Those, too, are easily available. You'll also need some 30-gauge (or larger) insulated wire. Stranded wire works best, and you should try to use a color-coded arangement. Finally, you will need a 2- or 3-amp line fuse and three or four switches. I used an eight-position PC-board-mounted switch.

With one exception, the card edge and card-edge-connector sockets are wired in parallel. Example: Contact "A" on the contact board is wired to contact "A" on each card-edge socket. Contact "B" is wired to contact "B" on each socket, etc. (See the diagram, Figure 2.) The only exception to that is contact No. 21, which is the +5 volt supply from the computer to the expander board. This contact is wired through a switch for each socket, and then to contact No. 21 of the socket. Thus, what is plugged into the socket will be powered up only when the switch is closed. Contacts "Z" and No. 22 are common ground.

Another alternative for those who are of an experimental turn of mind, would be the use of a rotary switch mounted to a small panel. You'd want to use a switch with the same number of contacts as there are switches on the boards, or ports on the boards, and wire to the rotary switch instead of having individual switches at each port. While this might appear to complicate the circuit a bit, it would result in up-front control of the ports. Make sure you use a non-shorting rotary switch for this application, and the rotary switch can then be mounted in a small separate plastic box of its own and placed either atop or alongside the computer. The added convenience that this affords would make it worth looking into.

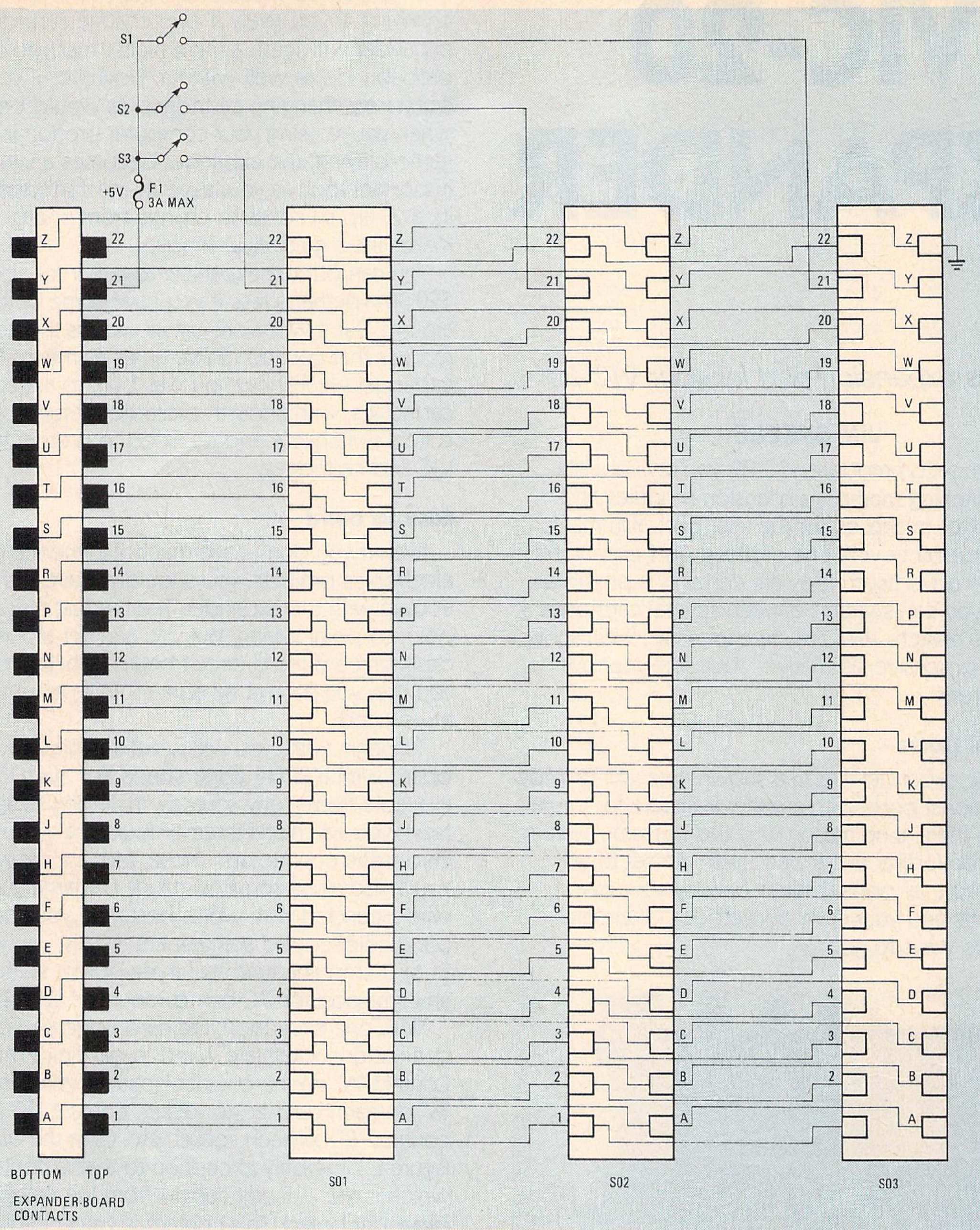


FIG. 2—SCHEMATIC DIAGRAM shows simple point-to-point wiring. Circuit is straightforward and direct with no hidden traps.

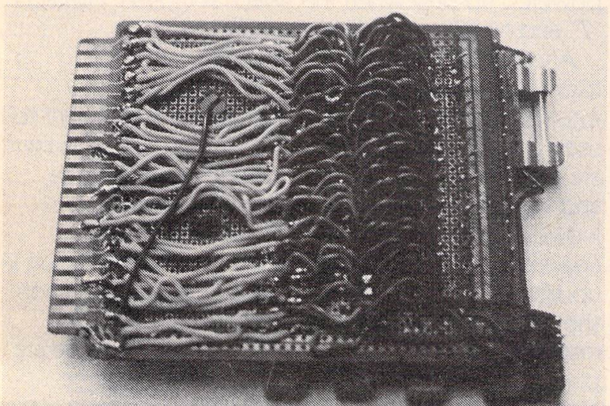


FIG. 3—UNDERSIDE VIEW REVEALS WIRING. The VIC-20 expander makes a good one-weekend project that will reward you with years of added convenience.

Check the wiring!

Once the wiring is completed, check the continuity of each circuit. This procedure is a *must* because any bad connection could cause your VIC to behave radically or crash memory at a most inopportune time. When you are sure that each connection is right, plug your unit into the VIC-20 expansion port, wire-side down, making sure the contacts line up. Plug in your game and/or your memory expansions. Check the operation of each of the expansion ports with a game you are familiar with or a memory expansion. If you turn on your VIC and it does not work properly, turn it off and recheck your wiring and make sure that the contacts from the plug line up with the expansion port contacts.

That's all there is to it! Unless you have more than three expansion modules and games you don't have to worry about plugging in a module every time you need one—just switch it on when you need it!