CIRCUIT CELLAR INK®

# THE COMPUTER APPLICATIONS JOURNAL

Computer-Generated Anaglyphs

High-Resolution Timing

MC68HC11 Powerhouse




December '91/January '92 — Issue 24

12
0 74470 75349 0

$3.95 U.S.
$4.95 Canada

# Our Upcoming Year

**EDITOR'S INK**

*Ken Davidson*

Now that I've scared off all our potential authors ("English: The Forgotten Language?" in issue #23), I want to share with you our editorial calendar for the upcoming year.

Issue #25 (February/March 1992)
Theme: Building Automation
Special Section: Embedded Signal Conditioning
Article Deadline: October 18, 1991

Issue #26 (April/May 1992)
Theme: Distributed Control
Special Section: Embedded Programming
Article Deadline: December 27, 1991

Issue #27 (June/July 1992)
Theme: Real-Time Programming
Special Section: Embedded Sensors and Storage
Article Deadline: February 28, 1992

Issue #28 (August/September 1992)
Theme: Signal Processing
Special Section: Embedded Interfacing
Article Deadline: April 24, 1992

Issue #29 (October/November 1992)
Theme: Measurement and Control
Special Section: Embedded Graphics and Video
Article Deadline: June 26, 1992

Issue #30 (December '92/January '93)
Theme: Debugging, Simulators, and Emulators
Special Section: Embedded Control and Conversion
Article Deadline: August 28, 1992

If you have an article idea related to one of the themes listed above, let us know about it. We'll work with you to develop your idea into a bang-up article. Even if you have an idea that doesn't fit one of the themes, we still want to talk to you. We run articles unrelated to the theme in each issue of the magazine and are constantly working on new volumes of The Circuit Cellar Project *File* books. As long as you know your stuff, we'll work with you to produce a literary wonder. Obviously, the deadline for issue #25 is past, and we are hard at work putting that issue together as you read this.

## THE ISSUE AT HAND

One of the most popular articles we've ever run was on generating holograms using a computer, some basic photography, and a laser ("Computer-Generated Holograms," CIRCUIT CELLAR INK, issue #14). The beauty of the concept was the simplicity of the code used to generated the images. Dale Nassar is at it again, producing more three-dimensional images using a computer, but this time with the same red/blue glasses with which you used to watch 3-D movies. As before, the code is very short, and the results are spectacular.

We also have another article from a popular author, Chris Ciarcia (Steve's brother, if you hadn't figured it out by now). This time, he's talking about designing optical systems (specifically, a lens) by computer. Most of our readers know all about circuit design by computer, but optical design is probably new to many.

Don't miss this issue's special section, where we present a nifty little controller that has everything you need for most control applications right on the board. The IBM PC keyboard interface and video output especially caught our eye.

## THE NEXT ISSUE

The theme of the next issue of CIRCUIT CELLAR INK is always a popular one: Building Automation. The Circuit Cellar engineering staff is hard at work designing the Home Control System II (HCS II) and will starting presenting the system's building blocks. We're also planning some changes and improvements to the look of the magazine I think you'll like.

Ken

FOUNDER/
EDITORIAL DIRECTOR
Steve Ciarcia

MANAGING EDITOR
Ken Davidson

ASSISTANT EDITOR
Lisa Nadile

ENGINEERING STAFF
Jeff Bachiochi
Ed Nisley

CONTRIBUTING
EDITORS
Tom Cantrell
Chris Ciarcia

NEW PRODUCTS
EDITOR
Harv Weiner

ART DIRECTOR
Lisa Ferry

PRODUCTION
MANAGER
Mark Vereb

STAFF RESEARCHERS:

Northeast
John Dybowski
Midwest
Jon Elson
Tim McDonough
West Coast
Frank Kuechmann

Circuit Cellar BBS—24 Hrs. 300/1200/2400 bps, 8 bii. no parity. 1 stop bii, (203) 871-1988; 9600 bps HST (203) 87 1 -0549

All programs and schematics in Circuit Cellar INK have been carefully reviewed to ensure that their performance is in accordance with the specifications described, and programs are posted on the Circuit Cellar BBS for electronic transfer by subscribers.

Circuit Cellar INK makes no warranties and assumes no responsibility or liibilii of any kind for errors in there programs or schematics or for the consequences of any such errors. Furthermore, because of the possible variation in the quality and condition of materials and workmanship of reader-assembled projects, Circuit Cellar INK disclaims any responsiblity for the safe and proper function of reader-assembled projects based upon or from plans, descriptions, or information published in Circuit Cellar INK.

Entire contents copyright © 1991 by Circuit Cellar Incorporated. All rights reserved. Reproduction of this publication in whole or in part without written consent from Circuit Cellar Inc. is prohibited.

Cover Illustration
by Robert Tinney

CIRCUIT CELLAR INK®

# THE COMPUTER APPLICATIONS JOURNAL

## In This Issue...

## FEATURES

16 **Three-Dimensional Graphics by Computer**
*Computer-Generated Anaglyphs*
by ***Dale Nassar***
Dale Nassar enjoys combining three-dimensional imaging with computers. Find out how to create images on your computer screen with realistic-looking depth.

30 **PC-Based Optical System Design**
*Using Your PC to Design a Lens*
***by Chris Ciarcia***
Most of us are familiarwith circuit simulation by computer, but how often do we consider doing the same for optical systems? Chris Ciarcia gives us a sampling by showing how to design a lens by computer.

38 **A Video Editing Control System-Part 2**
by ***William J. Kressbach***
In the last issue, Bill Kressbach described the hardware aspects of a video editing controller. In this final part, he gives the software particulars.

46 **High-Resolution Timing on a PC**
***by Bruce Ackerman***
Timing events with microsecond resolution using your IBM PC is easier than you think. See how it's done with no additional hardware.

60 **Third Design Contest Results**
***Winners of Circuit Cellar INK's Third Annual Design Contest***
The results are in. Yet again, the judges had a difficult time selecting from a superb field of entries. We salute this year's design contest winners.

## SPECIAL SECTION: Single-Board Solutions

52 **The FS- 100 MC68HC11 -Based Single-Board Computer**
*On-board Keyboard and Video Interfaces Complete* this *Single-Board Powerhouse*
***by Frank Swiger & Joe Glover***
Take the popular MC68HC11, add an IBM PC keyboard interface, a video output, and some additional I/O and you have a super single-board solution.

# DEPARTMENTS


1 **Editor's INK**
**Our Upcoming Year**
*by Ken Davidson*

4 **Reader's** INK-Letters to the Editor

10 *NEW*ProductNews

66 **Firmware Furnace**
**Micro-C, LEDs, and a Rotary Encoder**
*by Ed Nisley*

75 **From the Bench**
**Electronic Identification**
*by Jeff Bachiochi*

83 **Silicon Update**
**Summer Harvest**
*A Cornucopia of Chips*
*by Tom Cantrell*

91 **Practical Algorithms**
**Parallel Filters**
*Design and Simulation by Computer*
*by Charles P. Boegli*

98 **Domestic Automation**
**Home Automation: Read All About It**
*by Ken Davidson*

100 **ConnecTime**—Excerpts from the Circuit Cellar BBS
*Conducted by Ken Davidson*

112 **Steve's Own INK**
**Consultant's Dilemma**
*by Steve Ciarcia*

97 **Advertiser's Index**



PUBLISHER
*Daniel Rodrigues*

PUBLISHER'S ASSISTANT
*Susan McGill*

CIRCULATION COORDINATOR
*Rose Mansel/a*

CIRCULATION ASSISTANT
*Barbara Malleski*

CIRCULATION CONSULTANT
*Gregory Spitzfaden*

BUSINESS MANAGER
*Jeannette Walters*

ADVERTISING COORDINATOR
*Dan Gorsky*

**HAJAR ASSOCIATES**
**NATIONAL ADVERTISING REPRESENTATIVES**

NORTHEAST
**Debra Andersen**
(617) 769-8950
Fax: (617) 769-8982

MID-ATLANTIC
**Barbara Best**
(908) 741-7744
Fax: (908) 741-6823

SOUTHEAST
**Christa Collins**
(305) *966-3939*
*Fax:* (305) 985-8457

MIDWEST
**Nanette Traetow**
(708) 789-3080
Fax: (708) 789-3082

WEST COAST
**Barbara Jones & Shelley Rainey**
*(714)* 540-3554
Fax: **(714) 540-7103**

CIRCUIT CELLAR INK (ISSN 0896-8985) is published bimonthly by Circuit Cellar Incorporated, 4 Park street, suite 20, Vernon, CT 06066 (203) 875-2751. Second-class postage paid at Vernon. CT and additional offices. One-year (6 issues) subscription rate U.S.A. and possessions $17.95, Canada/Mexico $21.95, all other countries $32.95. All subscription orders payable in U.S. funds only, via international portal money order or check drawn on U.S. bank. Direct subscription orders to Circuit Cellar INK, Subscriptions, P.O. Box 3050-C, Southeastern, PA 19398 or call (215) 630-1914.

POSTMASTER: Please send address changes to Circuit Cellar INK, Circulation Dept., P.O. Box 3050-C, Southeastern, PA 19398.

# READER'S INK

# Letters to the Editor

## FINAL WORD ON STANDARD DEVIATION

Since the exchange letters between Charles Boegli and me ("Reader's INK," issues #21 and #22 of CIRCUIT CELLAR INK) regarding his article, "Adjusting Standard Deviation to Sample Size" ("Practical Algorithms," issue #20), I've done a bit of research to resolve the question of the $c_2$ adjustment divisor. (Yes, Norman, there is a $c_2$!)

Briefly, if the sample drawn from the population is unbiased, then the sample standard deviation,

$$s = \sqrt{\frac{\sum (x - X)^2}{n - 1}}$$

is the best estimate of s, the standard deviation of the population. A sample whose elements are drawn at random from the population is unbiased. An example would be a designed experiment wherein the experimental runs were done in random order.

On the other hand, in running a control chart on an ongoing process, s is usually estimated from a series of successive runs, rather than randomly. This sample is biased, since no data on future runs are used. In this situation the best estimate is s/c,, where $c_4$ corrects the bias. This is a larger estimate than for the random situation, and reflects the added uncertainty of the outcome of future runs. If the sample standard deviation is estimated by dividing the sum of squares by $n$ (a time-honored practice, but a bad one in my opinion), then the correction divisor is c,. The estimated value of s is the same by either method of calculation. The values of $c_2$ and $c_4$ are functions of the sample size, $n$. The exact expressions involve gamma functions (not commonly available on your handy pocket calculator), but can be approximated closely by a square root fit, as Mr. Boegli has neatly shown.

The bottom line is that the proper estimate to use depends on the sampling protocol. This point was not brought out in Mr. Boegli's article. I think it is important to do so.

Norman F. Stanley
Rockland, ME

## WHAT ABOUT THE SCHEMATIC READER?

I felt compelled to write after reading Bruce Webb's article "OrCAD Schematic Design Tools" (CIRCUIT CELLAR INK, issue #23). As a technician who has to use OrCAD-generated drawings, I'd like to pass on a few recommendations and observations to those folks who do the circuit layouts.

1) Signal names. Names cut down on repair time. Names need to be consistent, no name changes should be made on a signal path unless that signal is modified through a component.

My personal experience with poor naming technique is with a drawing of a CPU board that has several lines that go through up to six name changes without going through another part, then to top it off, the lines are not connected to anything.

2) Power and grounds. Granted, any engineer or technician should know where the power and ground pins are at on most 7400 TTL and CMOS chips. How about ECL? Or a proprietary chip? Or surface-mount PLCC? Or a PAL, Xilinx, or other related programmable chip?

The drawings that I've seen rarely show power and ground connections on the chip. Sometimes there is a chart that lists those points; that helps. Otherwise you need a stack of data books to hunt down some of the more elusive connections.

3) Block diagrams. Sometimes it's faster to lay out the schematic as a block diagram, just to show how the major parts of a system are tied together. This is especially useful when the system you're working on is several pages deep. Showing that U78 on page 2 is connected to U4 on page 32 via a line called ACK01 as a one-page block does simplify matters and speeds up diagnostics.

In using OrCAD-generated schematics, I have had to resort to redrawing the schematics of several boards to block diagrams, just to make the various boards easier to visualize. After making the block diagrams, my time spent in searching through all the various schematics has been cut down by about 75%.

4) Programmable gate arrays. Yes I know, you don't want your competition to know your circuits work by using a PAL or similar device. What about your service personnel? If your field service rates are $150 an hour and

your service people can't troubleshoot a board because they don't know how a signal is acted upon thorough a PAL, whose fault is that? And are you going to bill the customer for your failure to inform that service person?

I routinely deal with PALs and Field-Programmable Gate Arrays. It ain't easy troubleshooting one of those things if you don't have a program listing or even some sort of logic analyzer. Especially some like a Xilinx part where there can be over a hundred signal connections going in and out of the chip and you have no idea what the serial ROM is supposed to do.

5) Signal Flow. Just as proper names are needed for signal paths, showing which way signal paths come and go can be helpful too. This is especially true for programmable chips, were an I/O line needn't be unidirectional. Having the standard left-to-right diagram flow with all or most of the inputs on the left side and the outputs on the right helps make things consistent.

6) Circuit Board Parts Locations. Whenever possible, have the parts on the board labeled or least laid out on a page sequentially. Some of the boards that I have worked on have U1 at one end of the board, U2 through U4 aren't used, then U5 is way over on the other side of the board. This consumes a lot of time looking for a part when you have a high-parts-count board.

7) Parts Lists. Having a way to find the right part is vital, as well as the means of getting it quickly. Make sure that a parts list goes with each schematic drawn.

Now back to OrCAD. On a scale of 1 to 10, I give OrCAD a 2, mainly because a poorly made schematic is still better then no schematic at all. Now, I have never seen OrCAD in action at a workstation, I just use the end result. However, I've talked with engineers who do use OrCAD , and they readily admit that for schematic generation, OrCAD just doesn't cut it. OrCAD might be good for net lists and such, but for us technicians, it's the pits.

As you might of noticed for both OrCAD and Schema articles, the points that go off the illustrated schematic don't say anything about were they go. The diagrams I work with are up to 20 pages thick. I don't want to spend the time to hunt through 20 pages of drawings to find how parts are connected together. It's time consuming and frustrating trying to chase down a signal path when you have no idea where it's going to and coming from.

Schematic-generating programs, like word processors, do help in making information transfer easier. Engineers need to be aware of who will be using their drawings. Factory and field service people need all the information they can get their hands on to work with these circuits. Having the circuit information laid out in a logical and easy-to-follow way cuts down on service time, eases the technician's tension level, and helps to cut down, on overhead. My contention is that if written documents were generated the same way OrCAD drawings are, we would really be in a mess.

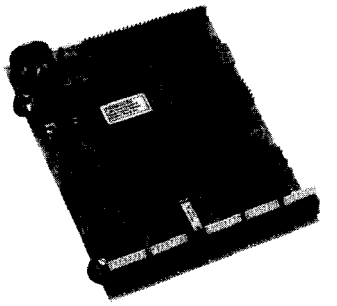
Thomas M. Nathe
Beaverton, OR


**Instant Microcontroller**

\+

**Instant C**

=



**Instant New Product**

Use our Little Giant™ and Tiny Giant™ miniature microprocessor-based computers to instantly computerize your product. Our miniature controllers feature built-in power supplies, digital I/O, serial I/O (RS232 / RS485), A/D converters (to 20 bits), solenoid drivers, time of day clock, battery backed memory, watchdog, field wiring connectors, and more! Designed to be easily integrated with your hardware and software. Priced from $159. Core modules as low as $59. Low cost, interactive Dynamic C™ makes serious software development easy.

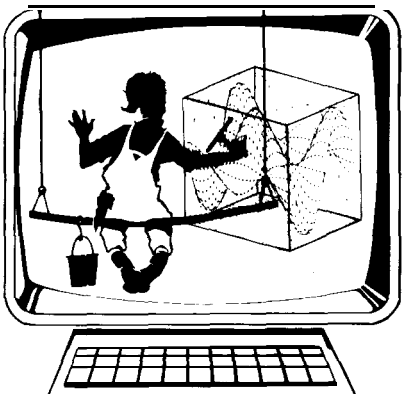
**Z-World Engineering**

1724 Picasso Ave.. Davis, CA 95616 USA
Tel: (916) 753-3722 Fax: (916) 753-5141
Automatic Fax: (916) 753-0618
(Call from your fax, request catalog #18)

Reader Service #211

**We do Windows**

***And Floors. And Walls.***



Use Plot-IT® to turn complicated ideas into comprehensive graphs Quickly and easily

- Over 60 graph types including 30
- Ultra sharp graphics in full color
- Perform spreadsheet tasks with Plot-IT® Worksheet
- User-customized FORTRAN functions support
- WordPerfect" and Ventura" compatible

And remember. we do Windows Microsoft Windows' operating system. that IS And IBM OS/2

For a tree brochure describing all the Plot-IT™ features. write or call now

Scientific Programming Enterprises
PO Box 669 • Haslett, Michigan 48840
(517) 339-9859 FAX (517) 339-4376

SPE SCIENTIFIC PROGRAMMING ENTERPRISES **Plot-IT®**

Reader Service #189

## TAKING ED TO TASK

Issue #23 arrived today! I went directly to my favorite: the "Firmware Furnace." What a shock to have Ed Nisley proclaim that he will no longer stoop to assembler ("C for microcontrollers has arrived") and then spends most of the rest of the column apologizing for lousy performance, unpredictable results, and incompatibilities across compilers. It will be interesting, you blinking lights with your Avocet and other megabuck compilers while your readers struggle ("...changing the startup code is not sinful-it can be essential") to figure out what has to be changed for our pedestrian compilers.

Face it, Ed. C sucks cycles, and is about as appropriate on an 8032 as COBOL was on a 4K 1401. For over 30 years I have programmed most of IBM's computers in all of the languages and C is the most illogical tripe ever foisted off onto "developers." I'll be damned if I'll fork over $600 to prove you can blink lights with C.

T.B. Kester
Atlanta, GA

*Ed Nisley responds:*

Give me one more chance! "Firmware Furnace" in this issue explores an 8031 C language development system that costs $100: Micro-C from Dave Dunfield. While not exactly free, it brings "real C" to the 8031 in a way that kilobuck compilers just can't touch. I've used it for several nontrivial projects and it's a winner.

But I agree with you: not only does C suck cycles, it bulges at the seams of the 8031's code space. That's why I don't hesitate to drop in huge blocks of assembler code where it will do the most good. C provides control logic and formatted I/O, leaving fast bit-banging to assembly code where it belongs.

For example, the firmware for an X-10 modem (which you'll read about in issue #25) weighs in at 900 lines of C and 700 lines of assembler. C was neither fast enough nor small enough for everything, but it let me concentrate on the hard parts and get the whole job done faster.

Ah, yes: the modem has three LEDs. The C code gets to blink only one of them...

### We Want To Hear From You!

Write letters of praise, condemnation, or suggestion to the editors of Circuit Cellar INK at:

Circuit **Cellar** INK
**Letters to the Editor**
**4** Park Street
Vernon, CT 06066

Circuit Cellar BBS: "editor"


µPLC®
LADDER LANGUAGE
AVAILABLE IN SOURCE CODE

The basic building block for your control system !

EXOR µPLC PROGRAMMING MODULE
LADDER LOGIC
RUN-TIME ENGINE
RS 232/485 (or PC I/O cards)
CPU's : 8051 Z80 8086
ANALOG I/O
DIGITAL I/O
YOUR EMBEDDED CONTROL SYSTEM

JAS : Call subroutines written in your favorite language. C, ASSEMBLER ....

The µPLC operating system kernel trasforms your bask microprocessor into a high- performance Programmable Logic Control.

Just plug the µPLC ROM into your favorite microprocessor card, load the integrated programmer/debugger onto your PC, connect a serial cable and begin taking the credit for a job well done!

Get started for only $ 399
Eval. Kit $ 99

- RS 485 multidrop distributed system
- Easy porting to custom hardware
- ROMable Run-Time Engine
- Programming tools
- On-line debugging
- Target library

*Eval. Kit contains :*
- *Parallel Port I/O or On Screen Simulator*
- *Programming & Debugging modules*
- *FC TSR Target*

*Excellent for ladder language training*

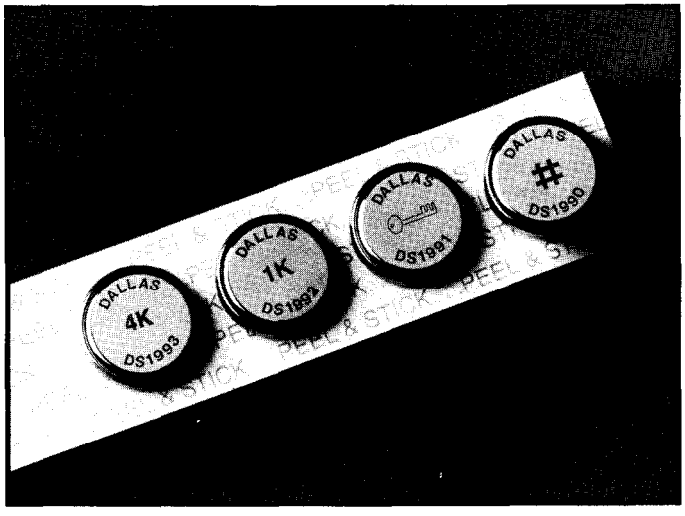
EXOR ELECTRONIC R & D

FAX (513) 874-3684 CALL (513) 874-4665
4850 Interstate Dr. Cincinnati, OH 45246

Reader Service #141

*conducted by Harv Weiner*

## TOUCH-CONTROLLED MEMORY CHIP




Dallas Semiconductor introduces DS199x Touch Memory, a nonvolatile memory chip in a durable, steel MicroCan. Touch Memory is a data carrier read and programmed by momentary contact with a computer. Available with adhesive backing, the Touch Memory firmly attaches data to virtually any object for instant availability.

Touch Memory shares some of the applications of a bar code, but it can be changed on demand. The chip can hold up-to-date, relevant information because it can be reprogrammed while attached. Available in densities of up to 4K bits, it records over 100 times the data of a bar code and transfers error-free data at a rate of 16K bits per second.

Touch Memory communicates through a one-wire signaling scheme. Special circuitry on the chip multiplexes address, data, and control wires onto a single bond pad. This bond pad is extended to the can's lid, and the rim and bottom become ground.

When contacted, the Touch Memory emits a wake-up signal, followed by its family code, a CRC code, and a unique 48-bit serial number. The serial numbers are written using a laser, and are registered and unalterable. The CRC code validates the serial number and qualifies the electrical connection.

The chip includes an internal scratchpad that prevents inadvertent writing over existing data or to the wrong location. Data is first written to the scratchpad and then verified before it is transferred to memory. Once the transfer is initiated, a copy of the scratchpad is faithfully reproduced in memory, even if the contact is broken.

Secure versions of Touch Memory require 64-bit passwords to be entered before data is transferred. In this way, mere possession of the chip is not sufficient to learn its contents.

Touch Memory can monitor product processing during manufacture, access control, asset management, and product information storage. Quickly retrofit any PC to read Touch Memory using the DS9097 COM port adapter. A model DS9092 Touch Probe is also available.

Prices for Touch Memory range from $1.58 to $3.71 in 1000-piece prices, depending on storage capacity. The DS9092K Touch Starter Kit, which provides hardware and software for quick evaluation of the Touch Memory family using a personal computer serial port, is available for $75.

**Dallas Semiconductor**
**4401 South Beltwood Pkwy.**
**Dallas, TX 75244-3219**
**(214) 450-0448**
Reader Service 1500

## PC CONTROL INTERFACE KITS

Home Control Concepts has simplified using a PC for home control applications with the introduction of two new devices. The Powerline Interface Kit allows PC control of lights, appliances, and other devices in the home by using X-10 modules (or compatible modules by Stanley, Heath, Tandy, etc.). The interface kit requires a PC with a serial or parallel port. Hardware and software installation takes about five minutes, and a software utility program verifies proper installation.

The Powerline Interface Kit aids in the development of custom "smart home" systems. Develop a system that uses the two-way interface to monitor the status of the home's lights and appliances, and make intelligent decisions based on their on/off status. Develop a home control system that uses IF-THEN logic. A Two-Way Powerline Interface Module (model TW523 by X-10 [USA] Inc.) is included with the kit that trasmits to and receives from the powerline. Programming consists of sending and receiving simple serial "packets" of information. In addition to the TW523, the interface kit includes cable, adapter, software (including a library of C utilities), documentation, and technical data. The kit sells for $79.

The PC-to-Infrared Interface Kit allows PC control of infrared devices in the home. The interface kit requires a PC with a serial port and a remote control (One-For-All or 1-Control). A software utility program called SendIR allows one to "push buttons" on the remote control from the computer's DOS environment. For example, a simple command from DOS can mute a stereo system or change a television channel.

The interface kit aids in the development of custom systems. Write software either around the SendIR utility program or from scratch with the sample C code provided. The PC-to-Infrared Interface Kit includes cable, hardware interface, software, documentation, and technical data, and it sells for $79.

**Home Control Concepts**
**9353C Activity Rd.**
**San Diego, CA 92126**
**(619) 693-8887**
Reader Service #501

## DSP SOFTWARE FOR IBM PC

Durham Technical Images has released PC Data Master 3.0, a Digital Signal Processing (DSP) software package for the IBM PC and compatibles. It offers a range of DSP and data display functions, an interactive help facility, and a pull-down menu interface. The basic PC Data Master distribution forms a complete system for DSP, but data analysis functions written by the end user are easily integrated into the package using virtually any language compiler or assembler compatible with MS-DOS 3.0 or higher.

The PC Data Master shell provides both pull-down menus and traditional DOS command windows for user interaction by either mouse or keyboard. Create multiple graphics windows for data display, and control them by mouse, shell commands, or dynamically from application routines. The interactive help utility is driven by a chain of help files, which can be expanded to describe user-written modules using any text editor.

Efficient binary data pipes are used to link processing modules implemented as independent executable (EXE) files. Process data files of any size using these pipes. If insufficient RAM is available, the shell automatically creates temporary spill files to buffer the excess data. Add processing modules using any language that supports the standard DOS file I/O system.

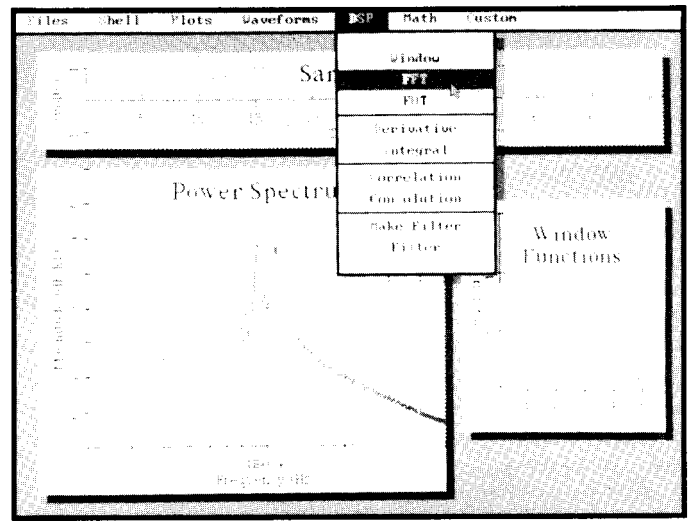
PC Data Master also has a set of DSP utility modules. These modules include fundamental operations such as forward and inverse FFT routines, convolution, correlation, window generation, FIR and IIR filter design and implementation, derivative and integral test data generation, and fifteen real and complex data file math functions. Implement many multistage transformations, such as the Cepstrum and the Hilbert Transform, without creating new software by combining these basic operations with data file math routines in data pipes.

The PC Data Master distribution includes a module for displaying waveform data. Use the plot system's autoconfiguration capabilities for quick display of individual or multiple data files. For more exacting applications, the plot system's interactive mode provides full or partial control over the design of data displays.

An add-on package is available to support two-dimensional signal processing. Called the "2D-Upgrade," this package extends the test data generation, data file math, FFT, convolution, and other modules in the basic package to handle two-dimensional data files. Graphics capabilities are also extended to include "waterfall" plots, "wire-mesh," 3-D surface plots, and contour plots.

PC Data Master sells for $185, including both the Basic Distribution and the Application Development Toolkit. An Academic Site License is available to universities for an additional $95. A Demonstration disk is available for $10, and the 2-D Upgrade costs an additional $85.

**Durham Technical Images**
**P.O. Box 72**
**Durham, NH 03824-0072**
**(603) 868-5774**
**Reader Service #502**




## DOWNLOADABLE EPROM EMULATOR

DPROM 2, a new version of a downloadable EPROM emulator, was introduced by Applied Data Systems. This unit handles larger EPROM sizes, has internal battery power for downloads, and has easier configuration.

DPROM 2 uses a microprocessor and RAM to simulate the operation of an EPROM in a target system. The unit uses a cable terminated in a DIP connector to connect to the target. An RS-232 connection transfers ROM code from a standard serial port. Because this emulator stores the ROM code in RAM, it allows changes to the code and overwrites of previous versions by downloading the revised file rather than burning EPROMS to test code revisions.

A single DRPOM 2 can emulate any 2732-27010 EPROM in an 8-bit system. Additional units may be daisy chained to fill multiple EPROM spaces, to provide emulation in systems with 16 or 32-bit buses, or to emulate a 27020 EPROM.

DPROM 2 has a NiCd battery to supply power during downloads, so connecting the unit to the target system during file transfer is unnecessary. The internal battery also backs up the memory contents. DPROM 2 operating parameters, such as baud rate and EPROM type, are easily set by means of a rotary switch and a push button.

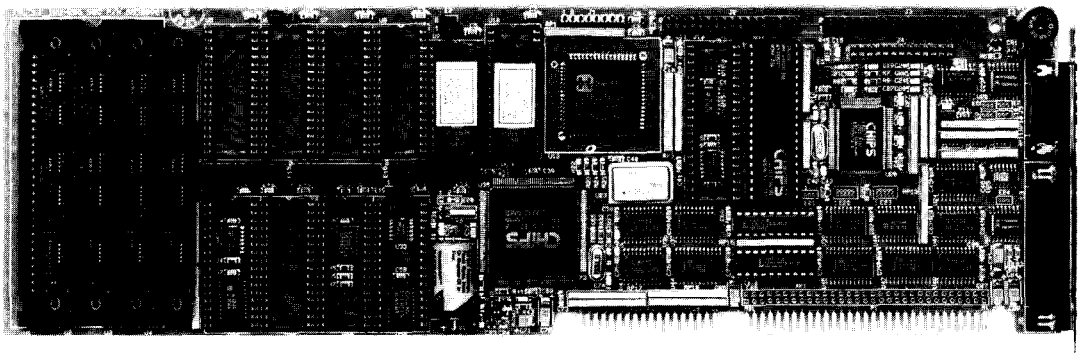
This emulator can receive file transfers at 9600, 19.2k, 38.4k, and 57.6k baud rates. It accepts files in Intel hex, Motorola S, and binary formats.

DPROM 2 is offered in four separate models. DP-64 with 64K of internal memory, which emulates EPROMs up to 27512. DP 128 with 128K of memory emulates smaller EPROMs as well as the 27010 and, when used in pairs, the 27020. Both models provide access time of 150 ns at the EPROM socket. The DP-64FM and DP-128FM models have the same capabilities, but offer access times of 100 ns at the EPROM socket.

The cost of DPROM 2 ranges between $125 and $250, with a thirtyday money-back guarantee.

**Applied Data Systems, Inc.**
**409A East Preston St. • Baltimore, MD 21202**
**(301) 576-0335 • Fax: (301) 576-0338**
**Reader Service #503**

## SINGLE-BOARD COMPUTER FOR EMBEDDED APPLICATIONS



IND-286 is a single-board computer for embedded systems applications with stand-alone diskless operation from Micro Computer Specialists Inc. It is a fully equipped 80C286 16-MHz, zero wait state, PC/AT-compatible computer on a standard size XT-type, plug-in board, specifically designed for backplane applications. IND-286 features ROM-DOS, a ROM-based, MS-DOS 3.3 compatible operating system that requires less memory and provides faster operation than MS-DOS.

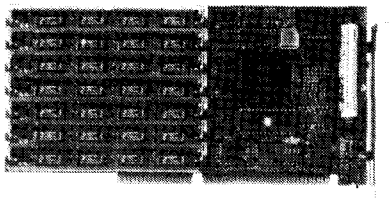
The board also includes a 4M-byte PROMDISK disk emulator with battery back-up and a watchdog timer. PROMDISK eliminates the need for a floppy-disk drive by allowing the user to boot the system and have the application software reside in flash EPROM or battery-backed SRAM. An on-board programmer for flash EPROMs simplifies field upgrades of applications software. This ability is crucial in embedded systems where the environments are too severe for mechanical disk drives. The watchdog timer facilitates using embedded applications for controlling critical processes where extended computer downtime cannot occur.

Additional features include up to 4 megabytes of DRAM, two RS-232 serial ports, a parallel printer port, a PC/AT-compatible keyboard port, a dual speed floppy-disk port, an IDE hard-disk port, a clock/calendar, and a socket for an optional 80C287 coprocessor. An embedded BIOS setup utility is included for system configuration and setting the clock calendar. An on-board I/O expansion connector interfaces with optional video daughterboards for applications requiring video. The power requirements for IND-286 are under 4 watts.

The price of the IND-286 computer is $795.

**Micro Computer Specialists, Inc.**
**25986 Fortune Way**
**Vista, CA 92083**
**(619) 598-2177**
**Fax: (619) 598-2450**
**Reader Service #504**


ROMDISK™



**High Performance Multimegabyte Disk Emulators**

**NEW MODELS / LOWER PRICES**

- Floppy Drive and multimegabyte emulators for ISA bus computers
- 180K to 14 MB capacities
- EPROM, Flash or SRAM technologies
- Autobooting, Single or Dual disk emulation under PC or MS DOS
- List prices from $195

CURTIS, INC.
**2837 No. Fairview Ave. • St. Paul, MN 55113**
**612/631-9512 FAX 612/631-9508**
PC DOS is a trademark of IBM; MS DOS is a trademark of Microsoft

Reader Service X131

A STEP BEYOND.




PROMICE takes ROM emulation a step beyond. It's an affordable, multi-operational development tool with:

- **on board intelligence**
- **modular design**
- **source level debugging**
- **future expandability**

*PROMICE. The Firmware Development System of Tomorrow...*

Reader Service #145

## COMPUTER TO VIDEO CONNECTION

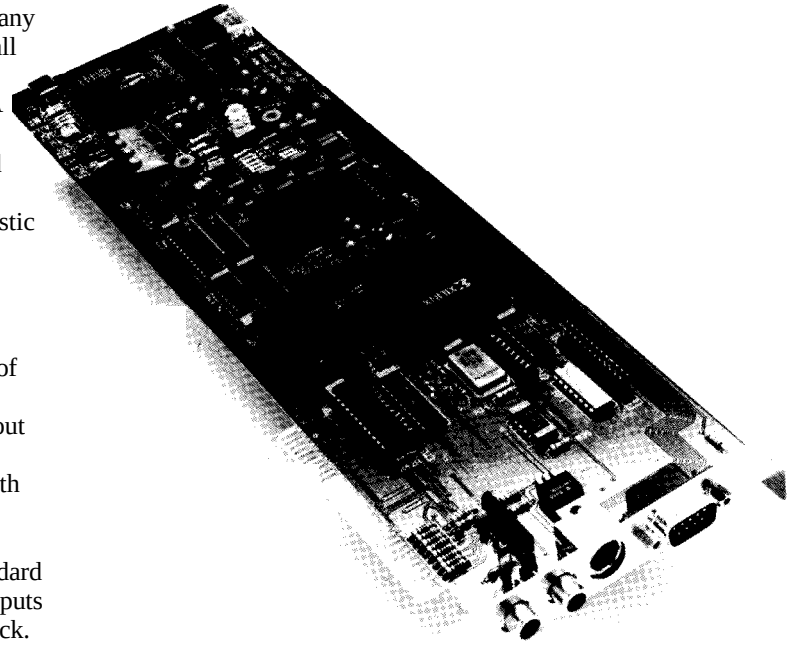
IEV International Inc. has released its VGA Broadcaster Board, which allows you to take what you create on a computer screen and record it on a VCR, display it on a TV, or project it with a video projector. The board connects to an existing VGA card feature connector and does not require any software reconfiguration or performance loss. It supports all standard VGA, EGA, CGA, and MCGA modes, and all extended VGA modes up to 768 x 480 x 256 colors.

The optional EDSUN Continuous Edge Graphics chip provides antialias operation for smoother edges, decreased "jaggies," and increased effective resolution for smooth images. With EDSUN, over 700,000 colors bring photorealistic capabilities to your output. Add-on software programs are available for animation, titles, and chalkboard. The Broadcaster is compatible with most software programs.

The Video Mixer mixes VGA graphics and video. By selecting any one of 256 colors, the simple superimposing of titles, credits, graphs, charts, maps, and so forth, creates a professional video output. Independent fade-in and fade-out control of both video and graphics are included.

The Broadcaster board comes as a complete system with software for digital sizing and positioning of video output. This board enables the user to "underscan" the computer screen image, so the complete image can be seen on a standard video monitor. NTSC, S-Video, and RGB RS-170 video outputs are provided, and the board accepts a sync input for Genlock.

**IEV International, Inc.**
3030 **South Main St., Suite 300 • Salt Lake City, UT 84115**
(801) 466-9093 • **Fax: (801)** 466-5921
**Reader Service #505**




TIRED OF WAITING FOR THE PROMPT ?

Speed up with a ROM DRIVE! Boots DOS and programs instantly. Also used to replace mechanical drive completely in controllers or diskless workstations. The only perfect protection from viruses. Easy to install half-size card.

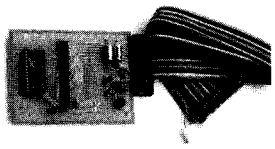
MVDISK1 64k.....$95
MVDISK2 360k....200
MVDISK3 1.44m...300

Quantity discounts!

**DOS IN ROM!**

**$95**





**$95 EPROM PROGRAMMER**

-PLUGS INTO PRINTER PORT
-DO ROMS FOR SBC OR MVDISK
-PROGRAM 2764/27010 EPROMS

WORLDS SMALLEST PC !!!

ROBOTS ALARMS RECORDERS DOS

THREE EASY STEPS:
1. Develop on PC
2. Download to SBC
3. Burn into EPROM

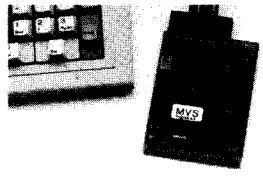
**$95**

**8088 SINGLE BOARD COMPUTER**

-2 PARALLEL -LCD INTERFACE
-3 SERIAL -KEYBOARD INPUT
-PC TYPE BUS -REAL TIME CLK
-BIOS OPTION -BATTERY OR 5V

FREE SHIPPING IN U.S.

5 YEAR LIMITED WARRANTY



MVS
Box 850
Merrimack, NH
(508) 792 9507

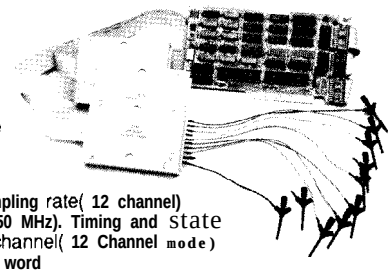
Reader Service #162

POWERFUL•AFFORDABLE INSTRUMENTS

**100 MHz Logic Analyzer**

**$799**
**LA12100**

Price is complete
Pods and Software
ncluded

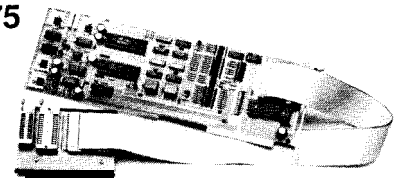


- 100MHz sampling rate( 12 channel)
- 24 Channels(50 MHz). Timing and state
- 2K samples/channel( 12 Channel mode)
- 24 Bit trigger word
- TTL threshold level
- Internal and External Clocks
- Menu driven software
- FREE software updates on BBS

• More sophisticated units also available

almost **UNIVERSAL PROGRAMMER**

PAL $475
GAL
EPROM
EEPROM
PROM
87xxx...

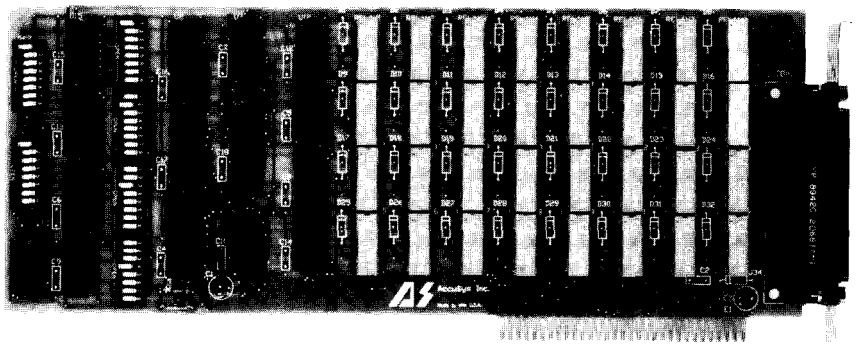


- 20 and 24 pin PALs,EPLDs
- 16V8 , 20V8 , 22V10 GALs
- 2716-27020 EPROMs
- 87xxx MICROs
- EEPROMs(incl 6 pin serial)
- Byte Split/Merge(16 & 32 bit)
- JEDEC, INTEL HEX, Motorola 'S' files
- Dallas NVS RAM programming
- PC/XT/AT COMPATIBLE
- FREE software updates on BBS

Call - (201) 808-8990
**Link Computer Graphics, Inc.**
369 Passaic Rd., Suite 100. Fairfield, NJ 07004 FAX: (201) 808-8786

Reader Service #157

## 32-SWITCH REED RELAY CARD



Achieve low-power data signal switching with any PC using the **32-Switch** Reed Relay Card from **AccuSys** Inc. The new card is an 8-bit device that selects and controls 32 analog or digital signals. The reed relays are completely transparent to the signals they are controlling, so any computer controlled signal up to 100 V and 10 W is unaffected. Easily manage low-power data signals such as telephone lines, temperature information, and flow rates. Use as many Switch Cards as the system has slots available; a PC has no address limit to the number of boards inside, so a single computer can control the connection of hundreds of signals.

The 32-Switch Reed Relay Card features four banks of independently addressable relay sets, and each port supplies eight sets of relays with a common connection to any signal or ground. Change port addresses with DIP switches from 280H to 727H. Control software primitives are supplied in assembler, BASIC, Pascal, C, FORTRAN, COBOL, and dBase. All address, data, and control signals are TTL compatible.

The reed relays are SPST dry reed devices with a life expectancy of $10^8$ contacts. The relay operating time is 0.5 ms and the breakdown voltage between open contacts is 250 VDC maximum, 1400 VDC maximum between coil and contact. The maximum contact current is 0.5 amp.

The card is XT, AT, and EISA slot compatible and is a standard full-size format. The 32-Switch Reed Relay Card sells for **$395,** software interface examples with source code included.

**AccuSys,** Inc.
3695 **Kings Row . Reno, NV 89503**
**(702) 746-I 111 . Fax: (702) 746-2306**
**Reader Service #506**


**EPROM PROGRAMMERS**

**Stand-Alone Gang Programmer** **$750.00**

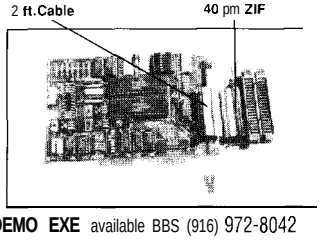



-Completely stand-alone or PC-driven
• Programs E(E)PROMs
• **1 Megabit of DRAM**
. User **upgradable to** 32 **Megabit**
• **3/.6" ZIF Sockets.,RS-232, Parallel In and Out**
• **32K** internal Flash EEPROM for easy filmware upgrades
• **Quick Pulse Algorithm 127256 in 5 sec, 1 Megabit in 17 sec.1**
• **2 year** warranty
-Made in the US A
• Technical support by, phone
-Complete manual and schematic
• **Single Socket Programmer also available. $550.00**
• Split and Shuffle 16 & 32 bit
• 100 User Definable Macros, 10 User Definable Configurations
. Intelligent Identifier
• Binary, Intel Hex. and Motorola S
• 2716 to 4 Megabit

**Internal Programmer for PC** **$139.95**

New Intelligent Averaging Algorithm Programs 64A in 10 sec 256 in 1 min 1Meg (27010. 011) in 2 min 45 sec., 2 Meg (27C2001) in 5 min Internal card with external 40 pin ZIF

. Reads Verifies and programs 2716.32 32A 64. 64A, 128 128A 256, 512, 513 010, 011.301, 27C2001, MCM 68764 2532, 4 Megabits
-Automatically **sets programming voltage**
• Load and save buffer to disk
• Binary, Intel Hex. and Motorola S formats
• No **personality modules rewired**
. 1Year warranty
• 10 days money back guarantee
. Adapters available for 8748, 49. 51, 751, 52. 55, TMS 7742, 27210, 57C1024, and memory cards
. Made in US A




**EMPDEMO EXE** available BBS (916) 972-8042

**NEEDHAM'S ELECTRONICS**
4539 Orange Grave Ave *Sacramento. CA 95841
(Monday-Friday, 8 am-5 pm PST)
C.O.D. MasterCard VISA

Call for more information
**(916) 924-8037**
**FAX** (916) **972** 9960

Reader Service #174

**PCBoards**

**P-C-B ARTWORK MADE EASY !**
Create and Revise **PCB's** in a Flash

* HERC, CGA, EGA, VGA, SUPER-VGA
* HELP SCREENS
* EXTREMELY USER FRIENDLY
*AUTO GROUND PLANES
* DOT- MATRIX, LASER and PLOTTER ART
* GERBER and EXCELLON OUTPUT
* CREATE YOUR OWN FILMS with **1 X** ART
*LIBRARIES
* DOWNLOAD DEMOS from **24** hr. BBS!

**REQUIREMENTS: IBM PC or Compatible, 384K RAM DOS 3.0 or later. IBM compatible printers, HP Laser**

**PCBoards -** layout program 99.00
**(PCBoards HP or HI PEN PLOTTER DRIVER 49.00)**
**PCRoute -** auto-router 99.00
**SuperCAD -** schematic pgm. **99.00**
Demo Pkg. - ( **includes all 3 programs) 10.00**

**Call or write for more information**
PCBoards
**2110 14th Ave. South, Birmingham, AL 35205**
1-800-473-PCBS / (205) 933-1122
BBS / FAX (205) 933-2954

Reader Service #180

# NEW PRODUCT NEWS

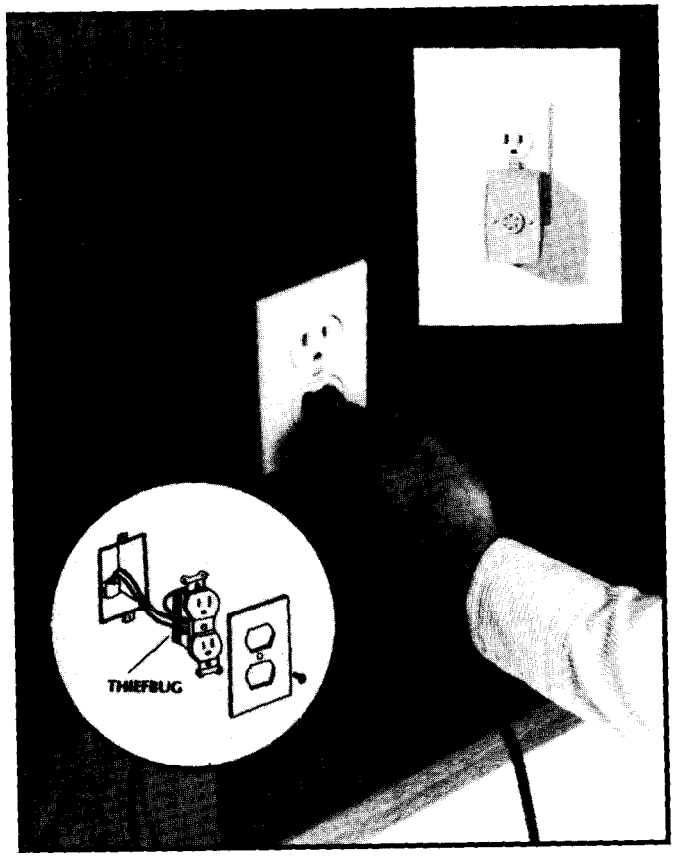
## CONCEALED THEFT MONITORING DEVICE

Thiefbug, presented by **CEPCO,** protects PCs, monitors, printers, TVs, VCRs, and other electrical items from theft or unauthorized removal. The device is installed in a standard electrical outlet box and functions as an ordinary outlet. When an item is disconnected, either by unplugging or cutting the power cord, the device transmits a coded alarm signal through the existing AC-power wiring to a monitoring unit at another location.

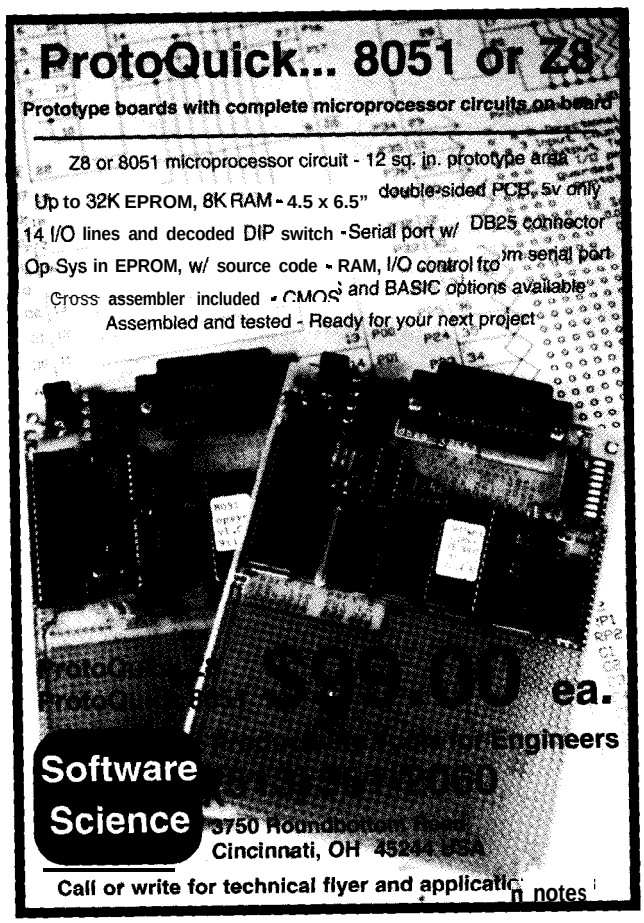
The monitoring units require no wiring and plug directly into the 115-VAC power outlet, monitoring as many Thiefbugs as needed. The monitoring units range from a single, plug-in buzzer or relay module to wall-mounted monitoring panels that identify up to 512 locations from the coded signal and provide audible, visual, and electrical (relay) outputs.

A complete system for protecting one PC (Thiefbug and buzzer or relay module) is priced at $83.

**CEPCO**
21515 Parthenia St.
**Canoga Park, CA 91304**
**(818) 998-7315**
**Reader Serivce #507**



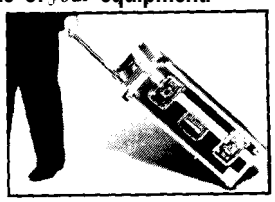




Reader Service #196

# Take It Easy.

**Take it easy on your cargo with a custom Cabbage Case built to the exact dimensions of your equipment.**



Take it easy on your back with our extension handle and tilt wheels options. Take it easy on your wallet. Let Cabbage Cases show you how easy it is to save money on quality, custom-built road cases that make shipping and traveling with your valuable cargo safer and easier. **Prices quoted over the phone.**
**Call 800-888-2495 today.**

**CABBAGE CASES, Inc.**
1166-C Steelwood Rd.
Columbus, OH 43212
**800/888-2495**
**614/486-2495**
**FAX/486-2788**

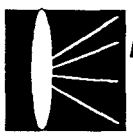
Reader Service #116

## FEATURE ARTICLES




page 16

**Three-Dimensional Graphics by Computer**



page 30

**PC-Based Optical System Design**



page 38

**A Video Editing Control System -Part 2**



page 46

**High-Resolution Timing on a PC**


# Three-Dimensional Graphics by Computer

## Computer-Generated *Anaglyphs*

Light: the miraculous carrier that transports colorful boundless information into our minds as we probe the world around us through a most remarkable pair of biological opto-isolators. With this built-in real world interface, the most glamorous element of the PC is its ever-improving video display.

In this article, I will describe a way to graphically simulate the depth of our real world on your PC's two-dimensional screen, using the color and resolution features of your video system. Because I am mocking part of the human visual process, I will look at it briefly. Most quantities that pertain to the human eye are approximations determined by formulas, definitions, and references. Although most calculations do not involve exact numbers, the results agree closely enough to be consistent with the principles applied in this article. Also, I'll describe what could become the basis of some very peculiar graphics applications, the number of which seems almost limitless.

You are able to see depth because each eye "sees" your world from a slightly different viewpoint and calculates the distances of objects using mental triangulation, a process more commonly known as depth perception. While these actions could be simulated with a computer in several ways, I will use a fun method: the old "red and blue 3-D glasses" or anaglyphic technique prevalent (albeit rarely) in specially filmed movies. Successfully incorporating this process to the PC requires the definition and plotting of two distinct, mathematically superimposed, color-coded perspective views of a scene. The colored eyeglasses separate the superimposed images and direct a particular image to the correct eye. This method seems realistic because it influences the brain to encode (in plotting), and subsequently decode (in viewing), the image in a manner that causes the eyes to refocus when shifting from a nearer to a farther part of the image, or vice versa. Compare this focusing sensation when you view the anaglyphs with that of standard "flat 3-D" computer images.

I chose the anaglyphic method because it produces beautiful images of strikingly realistic depth, and all you need are a couple pieces of inexpensive, colored plastic and your VGA. You can even program the colors to your taste or to whatever filter constructs you have lying around. Don't be discouraged by the quality of the 3-D movies you've seen. The computer-generated anaglyphs look much better. In fact, when you view the anaglyph, the depth is realistic enough that the image looks unrelated to the screen.

Fortunately, we're in the era of the low-cost high-powered VGA-based color video standards. When I started work on this project a few months ago, a standard VGA with sixteen colors at 640 x 480 pixel (pel) resolution had the highest quality per cost ratio for applications needing high-quality color graphics. Now the modes supporting any 256 of 256K colors at 1024 x 768 pel resolution are the best-and for the same or lower cost than the previous VGA standard. These high-color, high-resolution video modes open up a whole new

FEATURE ARTICLE

Dale Nassar

world of image synthesis possibilities. At this rate of improvement, I can't help but wonder what the next video breakthrough will bring.

## COLOR AND RESOLUTION: WHAT'S IDEAL?

**Now 1024 x 768** at **256** colors sounds great, and if you're familiar with the photographic quality obtainable with this super VGA graphics, you may wonder if improving the image quality further would result in significant improvement.

Consider aspects like pel density, which takes into account display size as well as pel resolution, when determining how much computer resolution the eye can resolve. For example, if you take a 10" x 7.5", 1024 x 768 graphic image and blow it up to the size of a giant TV screen, the image acquires a very grainy appearance. Although the resolution remains unchanged, the pel density lowers and the (local) image goes from high quality to poor quality. This change in appearance is important because of the great size differential between the PC screen and the minuscule retinal area of the human eye perceiving the significant visual information.

Let me compare the high-resolution graphics screen and the human eye, in terms of those resolution and color characteristics generated by the PC and generated by the eye. Very (very) loosely speaking, you "see" or perceive an image that is focused onto the retina by the lens of the eye. This optical information is "picked up" at the retinal surface by millions of receptor cells, which are connected by nerve fibers to the brain through the optic nerve. A portion of this image is focused onto the fovea, a tiny active central area of the retina. The information striking the fovea is responsible for sharp image perception, such as in reading and in concentrating on fine detail. The larger area outside the fovea is also light sensitive, but its role is mostly limited to peripheral vision.

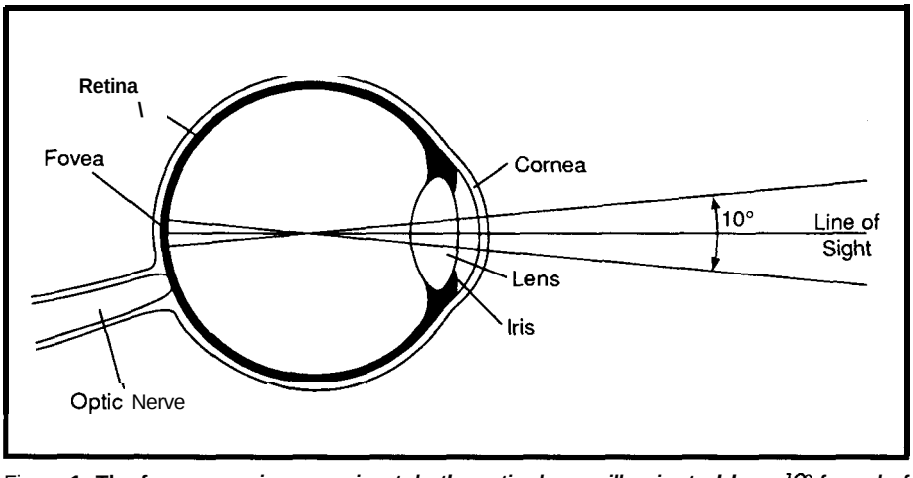



Figure **1 -The fovea** ***area is approximately the retinal area illuminated by a 10° funnel of light directly entering, and crossing at the center of. the eye.***

The key to this dual vision system is the two types of receptor cells lining the retina: canes and (smaller) **rods,** named according to their respective shape. Cones are primarily sensitive to bright light and are responsible for distinguishing colors. Rods cannot distinguish colors, but are more sensitive to light intensities, and are primarily active for night or low-light vision, This distribution of abilities is why colors are difficult to see in low light levels.

The fovea consists of a somewhat circular arrangement of tightly packed cone cells. Around the edge of the fovea, the cones become less packed and the rods more plentiful. This distribution explains why peripheral vision is better at night. In fact, directly viewing an object in low light is difficult because you focus on it with your fovea, which is relatively insensitive in low light. Strange as this fact may seem, focusing slightly away from an object brings the image information onto the rod cells for a better look under low-light conditions.

Obviously, the area of interest concerning my applications the most is the fovea. Therefore, I will concentrate on this small retinal region. Let me start by calculating its surface area, which is often defined in terms of funnel-shaped fields of vision entering the eye. As Figure 1 shows in cross section, the central field of vision is defined as the funnel-shaped volume of space bounded by a 10" arc emanating from the center (focal

point) of the sphere (eye) through the lens. Given the radius of the eye as 12.5 mm, I calculate the illuminated fovea area to be approximately 3.7 $mm^2$. In terms of resolution quantity, think of the fovea as a tiny PC screen only 1.9 $mm^2$. **The** fovea surface area has been approximated as slightly larger than a pinhead.

Now that you know the approximate retinal "screen size," let me describe resolution. The biological analogue of the pel is a retinal cone cell. The cone size (diameter) is also calculated as an arc similar to the fovea area calculation, but with a much smaller angle: 1/2 minute (0.00833"). The cone has a diameter of approximately 0.0032 mm and an area of about 0.00001 $mm^2$. Therefore, the corresponding pel density is approximately 300 pels/mm. For a 10" x 7.5", 1024 x 768 resolution VGA monitor, the pel density is about 4 pels/mm, not even close to that of the eye. Makes the eye look good doesn't it? But now let me calculate the actual number of pels involved in each device: for the fovea I get 600 x 600 = 360,000 pels. For the VGA screen 1024 x 768 = 786,432 pels, more than twice as many as the eye. Now are you relatively impressed by the VGA? Don't let the illusion fool you; examine all the angles and you will find the eye wins hands down!

You can easily determine if a graphical image could be made of a higher quality (or not) given viewer conditions. For example, suppose you want to view a 1024-pel, IO-inch VGA line at full eye resolution (300 pels/mm). Give the problem a little thought: the individual pels become more evident if you are closer to the screen and are obvious if you view them through a magnifier. Therefore, you must do the opposite: decrease the image size (without removing any pels) or move the image farther away to eliminate redundancy in the viewed image with respect to the resolution elements of the eye. Refer to the simple geometry depicted in Figure 1 and you'll see the line should be placed at a viewing distance of about 367 feet! This distance is the minimum the IO-inch line can be positioned from the eye while remaining in the 10" fovea arc. You are in no danger of exceeding your eye resolution while viewing even the most impressive super VGA graphics images.

While describing the resolution element of the eye as the size of three cones instead of one may be advantageous due to the three color types (red, green, and blue), given its high resolution, even this 300% resolution reduction is insignificant. Conversely, the pel density may actually be somewhat larger because the cones are compressed to about rod diameter in the fovea.

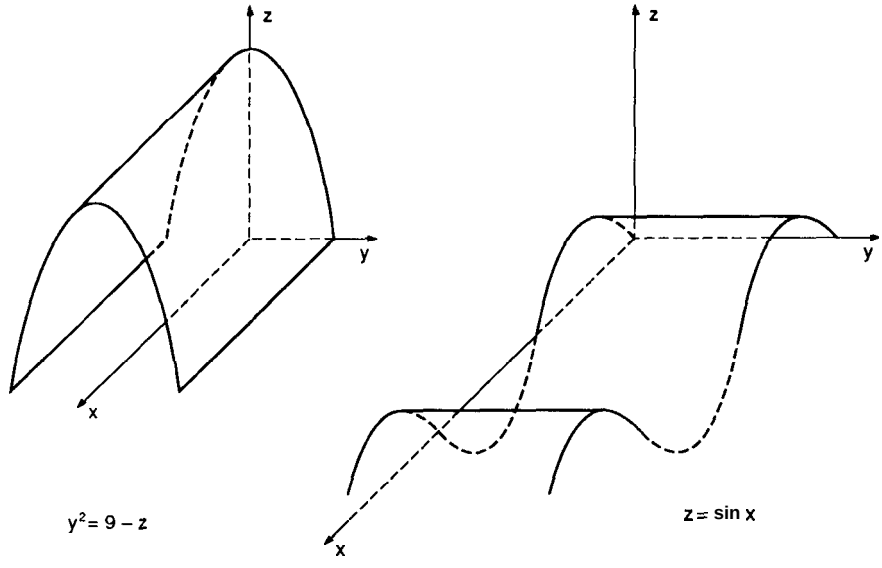



**Figure 2-Typical** *projections for causal viewing of three-dimensional surfaces.*

## ENCODING DEPTH ON THE SCREEN

Although the common PC has come a long way in display technology, it still lacks the characteristic most responsible for making the actual visual world around us realistic-depth.

To realistically simulate three-dimensional images using a less-than-ideal two-dimensional PC monitor, you must employ graphical projection techniques that take into account the nature of the image, the circumstances under which the viewer will observe the simulation, and the sensing characteristics of the viewing apparatus itself. In this case, as in most cases, the viewing apparatus will be a set of human eyes. The projection technique is anaglyphic as described earlier.

I chose rotated three-dimensional mathematical surfaces as the subjects of my original computer-generated anaglyphs. Many people would claim these are the most difficult to manipulate, but I feel the results are worth all the work.

The first step is to choose a suitable coordinate system. I chose a stan-

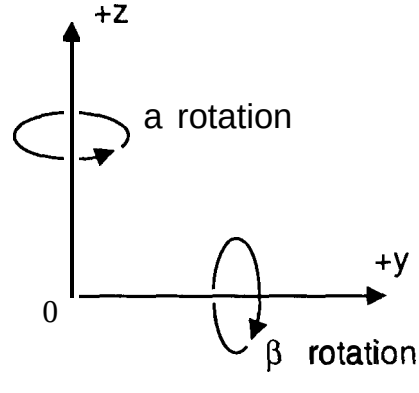


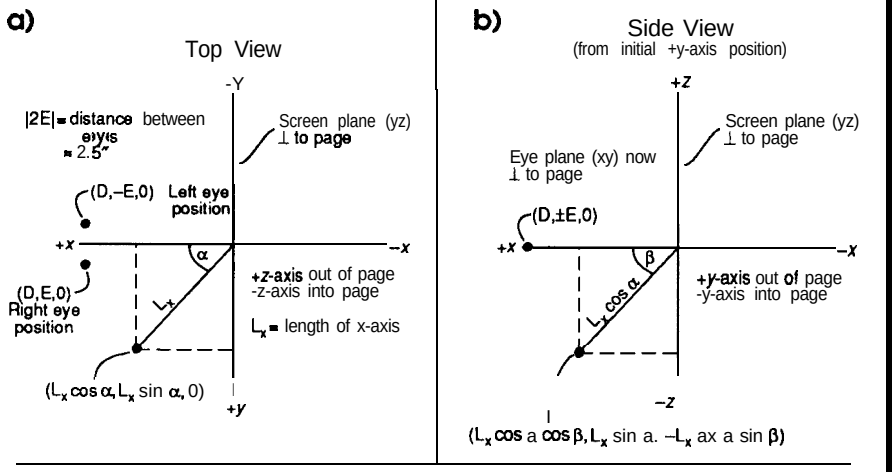

**Figure 3**—*Initially, the (unrotated) x-axis is perpendicular to the screen. The origin (0) projection always remains (pivots) at the* **center of the screen.** *The* **a rotation (rota-***tion about the z-axis) moves from the initial +x position toward the initial + y position. The β rotation (rotation about the y-axis) moves from the initial +z position toward the initial +x position. Negative rotations are simply 'evened.*

dard right-handed rectangular Cartesian coordinate system, consisting of three mutually perpendicular axes labeled x, y, and z. Figure 2 shows a typical system as represented by calculus and solid analytic geometry texts. While they are good for illustrative purposes, if a precisely true perspective is needed, these text illustrations won't suffice because they do have an inherent flaw. Note if the yz-plane is projected perpendicularly onto the page (as is the case), then the true x-axis should be perpendicular to the page and should have no projection. Because I need a mathematically accurate image definition, my projection technique cannot tolerate distortion. Also, the image must be rotatable if it is to be realistic as well as practical. Consider the plot of the top half of a sphere-a parachute shape. My view would be only a two-dimensional half circle if I plotted this surface with no rotation or tilting.

I also must have a method to realistically model a computer-generated surface form for my anaglyphs. Plotting all the pels making up the figure simply results in a solid half-circle with no clue of depth. I choose to use the "wire-frame" surface that represents the surface as a series of two sets of disjointed two-dimensional "slices" of the surface. One set is parallel to the x-axis and the other set is parallel to the y-axis, giving the surface a nice "netted" appearance.

My next step is to devise a method that accurately draws a set of rotated three-dimensional xyz-axes on the screen. This drawing is not just a parallel projection of the axes onto the screen, but an actual perspective projection given an actual viewing (eye) position.

Figure 3 shows the initial unrotated position of the axes. Note the origin is in the center of the screen with the initial position of the axes as follows: $+y$ is to the right, $+z$ is up, and $+x$ is perpendicularly out of screen. I will define two rotation angles: The first, alpha (a), is a clockwise rotation about the z-axis. Clockwise about the z-axis is defined as a rigid rotation about the z-axis such that the initial +x-axis rotates toward




**c)** Similarly, for $L_y$:

$$(-L_y \sin a \cos b + L_y \cos a + L_y \sin a \sin\beta)$$

is the y-axis endpoint position after a then $\beta$ rotations.

and $L_z$:

$$(L_z \sin\beta, 0, L_z \cos\beta)$$

For the $\alpha$-$\beta$ rotation of a general point $(x,y,z)$, we simply replace $(L_x,L_y,L_z)$ with $(x,y,z)$.

If we call our general $\alpha$-$\beta$ rotated points $(x_{\alpha\beta}, y_{\alpha\beta}, z_{\alpha\beta})$ then

$$x_{\alpha\beta} = x \cos\alpha \cos\beta - y \sin\alpha \cos\beta + z \sin\beta$$
$$y_{\alpha\beta} = x \sin a + y \cos a$$
$$z_{\alpha\beta} = -x \cos a \sin\beta + y \sin a \sin\beta + z \cos\beta$$

**d)**

$$x_L = D + (x_{\alpha\beta} - D)\, t$$
$$y_L = E + (Y, - E)\, t$$
$$z_L = z_{\alpha\beta} t$$

Equations shown here are for the right eye (E). For left eye, use -E.

Note that when $t=0, (x_L,y_L,z_L) = (D,E,0)$ which is the viewpoint. When $t=1, (x_L,y_L,z_L) = (x_{\alpha\beta},y_{\alpha\beta},z_{\alpha\beta})$.

The intersection of this line with the yzplane is the desired projection $(y_p,z_p)$ onto the screen.

The hidden line removal routine will take advantage of the fact that as $t$ varies from 0 to 1, the line point moves from the surface point to the eye (see text).

**e)**

$$y_P = E + (y_{\alpha\beta} - E)\, t$$
$$z_P = z_{\alpha\beta} t$$

where $t = D / (D - x_{\alpha\beta})$

To eliminate distortion when viewing with the naked eye, use parallel projections: letting $D \to \infty$, we get $t \approx 1$, therefore

$$y_P = y_{\alpha\beta}$$
$$z_P = z_{\alpha\beta}$$

for standard viewing.

**f)**

$$x_{\beta\alpha} = x \cos\alpha \cos\beta - y \sin\alpha + z \cos\alpha \sin\beta$$
$$y_{\beta\alpha} = x \sin a \cos\beta + y \cos a + z \sin a \sin\beta$$
$$z_{\beta\alpha} = -x \sin\beta + z \cos\beta$$

Figure 4—*(a)* **The x-axis** *end* **point position** *after a* **rotation.** *(b)* **The x-axis endpoint** ***position*** *after* $\alpha$ **thenprotation.** *(c)* **Ihe formula** *for* $\alpha$-$\beta$ *rotated position* $(x_{\alpha\beta}, y_{\alpha\beta}, z_{\alpha\beta})$ *of* $(x,y,z)$. *(d) Jhe* ***parametric*** *equations (parametert) ofaspace line through viewpoint* $(D,E,0)$ *androtated point* $(x_{\alpha\beta}, y_{\alpha\beta}, z_{\alpha\beta})$. *(e) The Intersection of the viewline* $(x_L, y_L, z_L)$ *with the screen (yz plane) is plotted as* $(y_p, z_p)$. *This gives the required perspective projection for the anaglyphic images. (f) The formula to rotate (x, y, z) in a* $\beta$ *then* **a** *sequence. This formula undoes* $\alpha$-$\beta$ *rotation.*

the initial +$y$-axis position. Negative rotations simply turn in the opposite direction. The α rotation allows you to look around the surface if you wish to picture one in this coordinate system. This rotation simulates walking around the object and is much like a "turn." The second rotation, beta (β), is a rotation about the initial $y$-axis position in which a positive rotation moves the initial +$z$-axis position toward the initial +$x$-axis position. As before, a negative rotation is a rotation in the opposite direction. This β rotation lets you "tilt" the surface. A +β rotation results in a downward tilting, letting you look at the top side of the surface. You can peek under it with –β rotation. Thus, with α-β rotations you can view a surface defined in this coordinate space from virtually any position.

## FORMULA DERIVATION

In the formula derivation process, I first derive a simple set of equations that, given an α rotation, a β rotation, and a viewpoint, simply calculates the endpoints of the three coordinate axes. I calculate and plot the intersection of the line through the viewpoint (eye) and the axis endpoint with the $yz$-plane because I want the perspective projection onto the $yz$-plane. The origin is always fixed at the screen center and is independent of rotation. Therefore, I simply draw straight lines

```
REM AXES PROJECTION———640x480x16

'*************************************************
REM 3D-AXES PROJECTION RED/GREEN/BLUE=X/Y/Z AXES
REM NEGATIVE AXES ARE DOTTED
REM ORIGIN (0,0,0) IS CENTER OF SCREEN
REM +Y IS TO RIGHT  +Z IS UP  +X IS OUT OF SCREEN
'*************************************************

CLS : PI = 3.1416: SCALE = 480 / 640: AXESLEN = 75
INPUT "Alpha rotation "; A
INPUT "Beta rotation "; B
INPUT "Viewing distance "; D
INPUT "1/2 eye seperation "; E

A = A * PI / 180: B = B * PI / 180: C = C * PI / 180 'DEG TO RAD
SCREEN 12
WINDOW (-100, -100)-(100, 100)

COLOR 4: LOCATE 26, 16: PRINT "X=RED"
X = AXESLEN: Y = 0: Z = 0
GOSUB 100
IF XAB < 0 THEN STYL = &HF0F0 ELSE STYL = &HFFFF
LINE (0, 0)-(YP, ZP), , , STYL
X = -50

COLOR 2: LOCATE 26, 36: PRINT "Y=GREEN"
X = 0: Y = AXESLEN: Z = 0
GOSUB 100
IF XAB < 0 THEN STYL = &HF0F0 ELSE STYL = &HFFFF
LINE (0, 0)-(YP, ZP), , , STYL

COLOR 1: LOCATE 26, 60: PRINT "Z=BLUE"
X = 0: Y = 0: Z = AXESLEN
GOSUB 100
IF XAB < 0 THEN STYL = &HF0F0 ELSE STYL = &HFFFF
LINE (0, 0)-(YP, ZP), , , STYL

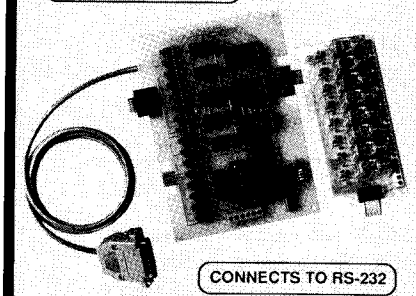
COLOR 5: LOCATE 27, 25: PRINT "Broken lines are into screen"
DO: LOOP WHILE INKEY$ = ""
COLOR 7
END

100 XAB = X * COS(A) * COS(B) - Y * SIN(A) * COS(B) + Z * SIN(B)
    YAB = (X * SIN(A) + Y * COS(A))
    ZAB = -X * COS(A) * SIN(B) + Y * SIN(A) * SIN(B) + Z * COS(B)
    T = D / (D - XAB)
    YP = SCALE * (E + (YAB - E) * T)
    ZP = ZAB * T
RETURN
END
```

**Listing 1**—*This program draws equal length x-, y-, and z-axes in red, green, and blue given an α rotation, a β rotation, half the distance between the eyes (E), and viewing distance (D).*


RELAY INTERFACE

PROVIDES SOFTWARE CONTROL OF RELAYS

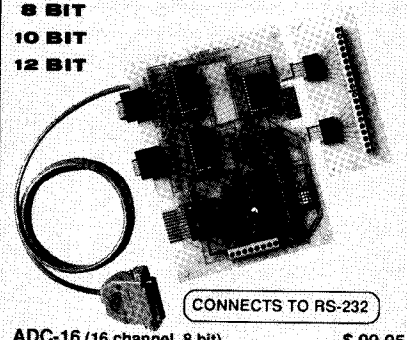


CONNECTS TO RS-232

**AR-16 RELAY INTERFACE**.....$ 89.95
Two 8 channel relay output ports are provided for control of up to 16 relays (expandable to 128 relays using EX-16 expansion cards). Each relay output port connects to a relay card or terminal block. A variety of relay cards and relays are stocked. Call for more info. RS-422 available (distances to 4,000 feet). PS-8 port selector may be used to control satellite AR-16 interfaces (up to 16,384 relays).
**RD-8 REED RELAY CARD (8 relays)**.....$ 49.95
**RH-8 RELAY CARD (10 amp SPDT 277 VAC)**....$ 69.95
**EX-16 RELAY EXPAN. CARD (16 channel)**.....$ 59.95

ANALOG TO DIGITAL

8 BIT
10 BIT
12 BIT



CONNECTS TO RS-232

**ADC-16 (16 channel, 8 bit)**.....$ 99.95
Input temperature, voltage, amperage, pressure, energy usage, energy demand, light levels, joystick movement and a wide variety of other types of analog signals. Inputs may be expanded to 32 analog or 128 status inputs using the AD-16 or ST-32 expansion cards. 112 relays may be controlled using EX-16 expansion cards. Analog inputs may be configured for temperature input using the TE-8 temperature input conversion. RS-422 available. PS-8 port selector may be used to connect satellite ADC-16 interfaces (up to 4,096 analog inputs/16,384 status inputs and 14,336 relays). Call for info on 10 & 12 bit converters. (terminal block and cable sold separately)
**ST-32 STATUS EXPAN. CARD (32 channel)**....$ 79.95
Input on/off status of relays, switches, HVAC equipment, thermostats, security devices, smoke detectors and thousands of other devices. (opto isolators sold separately)
**TE-8 TEMPERATURE INPUT CONV**.....$ 49.95
Includes 8 linear IC temperature sensors. Temperature range is minus 78 to 146 degrees F.
**TOUCH TONE DECODER** and other serial interfacing products available. Call for free information packet.

- **FULL TECHNICAL SUPPORT**...Provided over the telephone by our staff. EACH ORDER INCLUDES A FREE DISK WITH PROGRAMMING EXAMPLES IN BASIC, C AND ASSEMBLY LANGUAGE. A detailed technical reference manual is also included.
- **HIGH RELIABILITY**...engineered for continuous 24 hour industrial applications. All ICs socketed.
- Use with IBM and compatibles, Tandy, Apple and most other computers with RS-232 or RS-422 ports. All standard baud rates and protocols may be used (50 to 19,200 baud).
- Use our 800 number to order FREE INFORMATION PACKET. Technical information (614) 464-4470.

**24 HOUR ORDER LINE (800) 842-7714**
Visa-Mastercard-American Express-COD

**ELECTRONIC ENERGY CONTROL, INC.**
380 South Fifth Street, Suite 604
Columbus, Ohio 43215

Reader Service #136

E = half eye separation
D = viewing distance
A = α rotation
B = β rotation
XI = lower *x* boundary
XF = upper *x* boundary
DX = separation of *x* slices
YI = lower *y* boundary
YF = upper *y* boundary
DY = separation of *y* slices

AMP = amplitude of splash
DEC = decay of splash
FREQ = frequency of splash

(A and B units are degrees, all others are pixels)

For a quick cross-section view of an unrotated curve, try something like:

E,D,A,B,XI,XF,DX,YI,YF,DY = 0,9999,0,0,0,1,1,-300,300,3

A good starting reference for AMP, DEC, and FREQ is 200, 0.00002, 0.025

For a clean closed surface, XI and XF should be divisable by DX and YI and YF should be divisable by DY.

A math coprocessor is almost a must when experimenting with full surfaces.

**Figure 5**—*You should use the parameters described here as a starting point. Feel free to experiment to achieve different results.*

```
REM    SURFACE OF REVOLUTION (MATH SPLASH)  -----  1024x768
'$INCLUDE: 'gxlib.bas'
'$INCLUDE: 'grlib.bas'
retcode% = gxSetDisplay%(gxTS.38)
retcode% = gxSetMode%(gxTEXT)
PI = 3.1416
INPUT "E,D,A,B,XI,XF,DX,YI,YF,DY "; E, D, A, B, XI, XF, DX,
      YI, YF, DY
INPUT "Amp, Dec, Freq "; Amp, Dec, Freq
A = PI * A / 180: B = PI * B / 180
COSa = COS(A): COSb = COS(B): SINa = SIN(A): SINb = SIN(B)
COSaCOSb = COS(A) * COS(B): SINaCOSb = SIN(A) * COS(B)
COSaSINb = COS(A) * SIN(B): SINaSINb = SIN(A) * SIN(B)

retcode% = gxSetMode%(gxGRAPHICS)
retcode% = grSetViewPort%(0, 0, 1023, 767)
retcode% = grSetViewWorld%(-511, -383, 511, 383)
retcode% = grSetWorld%(gxTRUE)
retcode% = grClearViewPort%

FOR mesh = 1 TO 2
FOR x = XI TO XF STEP DX
XSQ = x * x
FOR y = YI TO YF STEP DY
YSQ = y * y: XSQYSQ = XSQ + YSQ

Z1 = Amp * EXP(-(Dec * (XSQYSQ)))
Z2 = COS(Freq * SQR(XSQYSQ))
Z = Z1 * Z2

XAB = x * COSaCOSb - y * SINaCOSb + Z * SINb
YAB = x * SINa + y * COSa
ZAB = -x * COSaSINb + y * SINaSINb + Z * COSb

TL = D / (D - XAB)
YP = CINT(E + ((YAB - E) * TL))
ZP = CINT(ZAB * TL)

retcode% = grPutPixel%(YP, ZP, grRED)
NEXT y
NEXT x
SWAP DX, DY
NEXT mesh
DO: LOOP WHILE INKEY$ = ""
```

**Listing 2**—*Before creating a complete anaglyph, it is useful to experiment with different parameters. This program displays a solid surface given the parameters described in Figure 5.*

from the center of the screen to each of the three projected axes endpoints (connecting two projected points gives the projected line), and I then have a perspective projection of the rotated coordinate system from only three plotted points. The beauty of this method is I have the complete rotation transformation equations for plotting any point with no more effort than drawing the axes if I let the endpoints of the coordinate axes be variable—sweet and simple.

Consider how a point is plotted in three-dimensional space: go along the *x*-axis to the desired value of *x*, then along the *y*-axis for the value of *y*, similarly for the z-axis, and then plot this final position. For example, to plot the projection of the point (3,1,2), calculate the two-dimensional (horizontal and vertical) projections onto the *yz*-plane of the, possibly rotated, points (3,0,0), (0,1,0), and (0,0,2). Then the final horizontal and vertical coordinates of the projection are the sum of the horizontal components of the three points and the sum of the vertical projections of the three points, respectively. To illustrate, suppose the two-dimensional *yz*-plane projection coordinates of (3,0,0), (0,1,0), and (0,0,2) are $(y_1,z_1)$, $(y_2,z_2)$, and $(y_3,z_3)$, respectively. Then the desired projection of (3,1,2) onto the screen is $(y_1+y_2+y_3,\ z_1+z_2+z_3)$. I find doing all my own derivations is much easier because the formulas used for my particular (and sometimes general) applications have no gray areas and they always work. In Figure 4, I detail the derivation with sketches and equations.

Listing 1 draws equal length *x*-, *y*-, and z-axes in red, green, and blue, respectively, given an α rotation, a β rotation, half the distance between the eyes (*E*), and a viewing distance (*D*). *[**Editor's Note:** Software for this article is available from the Circuit Cellar BBS and on Software On Disk #24. See page 105 for downloading and ordering information.]* Any axis that rotates behind the screen is dotted to ensure I see the correct perspective, eliminating the into or out of screen confusion that can occur with naked-eye viewing. Of course, if I pay attention to

my angle input I would know where everything should be and avoid any confusion. Run the program in Listing 1 with various *A* and *B* inputs (in degrees) until you fully understand the rotations. Be sure to do some negative rotations. For these trial runs, use *E = 0* and a large value for the viewing distance, *D* = 99999 for example. A large viewing distance simulates a parallel projection, which is more desirable in the nonanaglyphic case. Setting E to zero simply places the viewpoint on the initial x-axis line giving a straight-ahead view. Note all units are in pels. An example of a pel density calculation may be helpful: for a 640 x 480, 10" x 7.5" VGA screen you have 64 (640/10 or 480/7.5) pels/inch. A horizontal or a vertical scale factor may be used to equalize these two ratios if they are unequal on your screen. As I will explain later, using units of pixels in the drawings greatly simplifies some sticky points encountered in hidden-line removal procedures.

I am now in a good position to illustrate the difference between parallel and perspective projections using Listing 1. Parallel projection can be simulated by assigning a large viewing distance (*D*), which gives uniform image projections over the entire screen. For a smaller value of *D you* get a close-up view magnifying the nearer parts of the surface. This value can give a distorted appearance if the image is viewed with the naked eye. Try various viewing distances using a pel density of 64 per inch. Thus, for a viewing distance of 24 inches use a value of *E* = 1536. Notice what happens as *D* gets very near or even falls inside the axes space. You can even use negative *D* values that try to view the image from behind. Eye separation is typically 2.5 inches, so the standard value of E is 80 (1.25 inches). A nonzero value used for E moves the viewpoint off of the x-axis (it never moves out of the *xy*-plane). After this movement, the projected image corresponds to the viewpoint at *E*. Try various values of *E*, in conjunction with *D* if desired, until you fully understand its significance (don't forget the negative values).

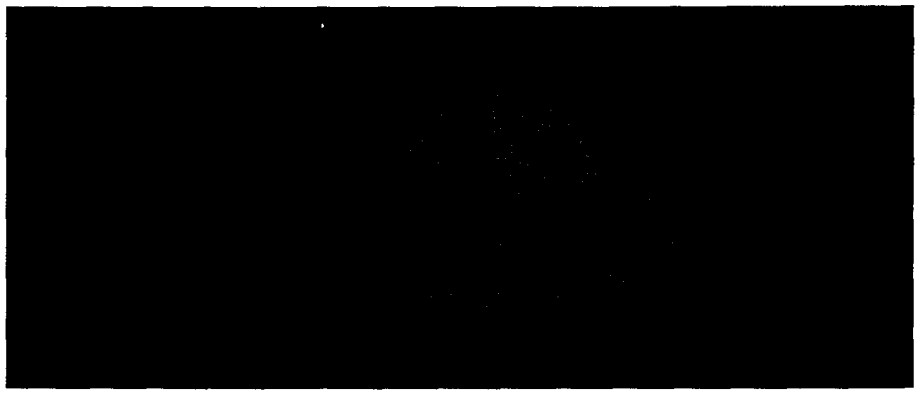


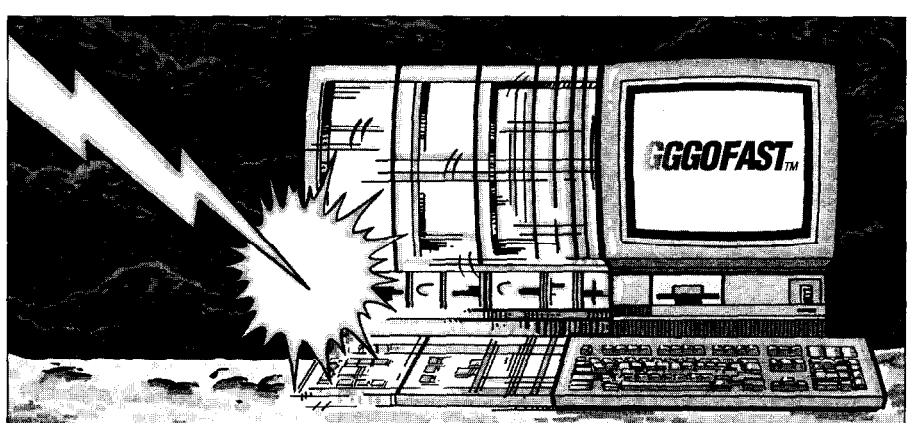
**Photo 1 --The *resulting display when the program in Listing 2 is run with the parameters in Figure 5.***

Note also when *E* is nonzero the projected z-axis now becomes tilted, which is impossible through simple α-β rotations. These features indicate the power of this little program. Using two values of *E* (±80) provides the two viewpoints of each eye in anaglyphic images; therefore, two separate (color-coded) images of the anaglyph are drawn.

A more mathematical way of getting parallel projections is with limits. Simply take the limit of the perspective projections as *D* approaches infinity and out pops the exact parallel projection formulas ideal for undistorted viewing of nonanaglyphic images.

## PLOTTING THREE-DIMENSIONAL MATHEMATICAL SURFACES

To illustrate this process and a couple of tricks involved, I'll make an interesting figure: the famous water-splash surface. Because I can't just guess at the equations that become great surfaces, I'll begin by defining a


GOFAST™

*Lightning-Fast Floating Point Accelerators*

Empower your 80x86 applications with GOFAST accelerators. Fast, reentrant, and ROMable, GOFAST accelerators boost performance and make sure you can embed your application.

Link and go with C: Microsoft@, Borland@, Intel®, MetaWare®, and WATCOM®. Dynamically replace 80x87 coprocessors during execution.

GOFAST IEEE accelerators are optimized for 8051, 8096, 8086, 80386, i960, 6801, 6809, 68HC11, 68xxx, 8085, Z80, R3000 and more.

Call for your free GOFAST information diskette today: 503-641-8446; FAX 503-644-2413; 800-356-7097.

© 1991 US Software Corporation. GOFAST and DynaCOP are trademarks of US Software Corporation. All other trademarks belong to their respective owners.

142 15 NW Science Park Drive
Portland, OR 97229

U S SOFTWARE®

Reader Service #203

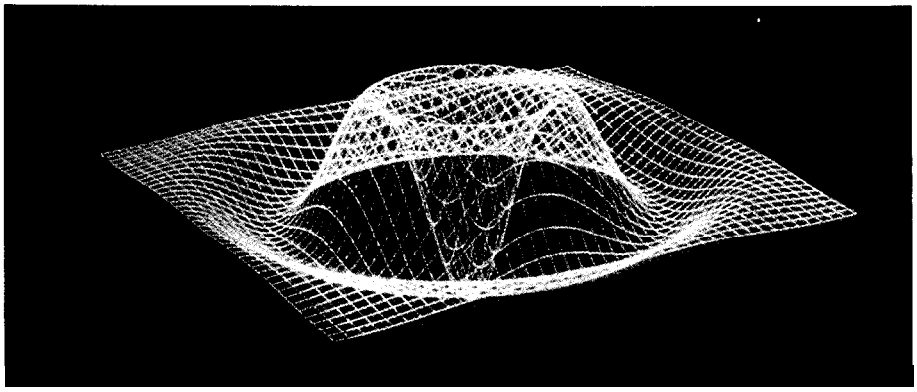


Photo 2-When **the *program in Listing 3 is run, this anaglyph is the result.***

two-dimensional surface and revolve or spin it about the z-axis, then I'll plot the rotated wire-frame model of the resulting "solid of revolution." These actions may sound a little difficult but they're not. I begin with a preview of the final surface. the two-dimensional curve that when revolved produces the desired surface using a cosine wave with an exponentially decaying amplitude. Figure 5 shows how this curve is created. If you use this method, getting nice expected images on the screen on the first try at a new surface becomes a snap. Otherwise, unless you know exactly what the equations will do and have a suitable scale, you will probably end up with a lonely pel or two plotted on the screen for your "finished image." Use Listing 2 to experiment with the parameters described in Figure 5 until you get a desirable preview.

```
REM TRANSPARENT ANAGLYPH--------1024x768
'$INCLUDE: 'gxlib.bas'
 'SINCLUDE:  'grlib.bas'
RETCODE% = gxSetDisplay%(gxTS.38)
RETCODE% = gxSetMode%(gxTEXT)
PI = 3.1416
INPUT "E,D,A,B,XI,XF,DX,YI,YF,DY "; E, D, A, B, XI, XF, DX, YI,
      YF, DY
INPUT "Amp, Dec, Freq "; TI, U, V
A = PI * A / 180: B = PI * B / 180
cosa = COS(A): cosb = COS(B): sina = SIN(A  : sinb = SIN(B)
RETCODE% = gxSetMode%(gxGRAPHICS)
RETCODE% = grSetViewPort%(0, 0, 1023, 767)
RETCODE% = grSetViewWorld%(-511, -383, 511  383)
RETCODE% = grSetWorld%(gxTRUE)
RETCODE% = grClearViewPort%

FOR image = 1 TO 2
FOR mesh = 1 TO 2
FOR X = XI TO XF STEP DX: XSQ = X * X
FOR Y = YI TO YF STEP DY: YSQ = Y * Y

Z1 = TI  * (EXP(-(U * ((XSQ) + (YSQ)))))
Z2 = COS(V *  (SQR((XSQ) + (YSQ))))
Z = Z1 * Z2

XAB = X * cosa * cosb - Y * sina * cosb + Z * sinb
YAB = X * sina + Y * cosa
ZAB = -X * cosa * sinb t Y * sina * sinb t Z * cosb

TL = D / (D - XAB)
YP  = CINT(E t ((YAB - E) * TL))
ZP = CINT(ZAB * TL)

pcolor% = grGetPixel(YP, ZP)
IF image = 1 THEN
    RETCODE% = grPutPixel%(YP, ZP, grRED)
ELSEIF image = 2 AND (pcolor% = grRED OR pcolor% = grMAGENTA) THEN
    RETCODE% = grPutPixel%(YP, ZP, grMAGENTA)
ELSE
    RETCODE% = grPutPixel%(YP, ZP, grBLUE)
END IF

NEXT Y, X
SWAP DX, DY
NEXT mesh
E = -E
NEXT image
BEEP
DO: LOOP WHILE INKEY$ = ""
RETCODE% = gxSetMode%(gxTEXT)
END
```

**listing *J-Adding a bit of extra code to Listing 2 results* in a pair of surfaces, one intended for each eye.**

I now have a nice flat curve, but how do I get a formula for the three-dimensional surface of revolution for a given two-dimensional curve? Simplicity itself. Starting with a standard two-dimensional function

$$y = f(x)$$

replace $x$ with

$$\sqrt{x^2+y^2}$$

and replace y with $z$. That's all there is to it! You end up with

$$z = f\left(\sqrt{x^2+y^2}\right)$$

Now plug in the desired rotation angles and surface parameters and plot the wire-frame model. The resulting screen image is shown in Photo 1.

## TRANSPARENT ANAGLYPHS

**Now** I'll construct an anaglyphic image of the water splash of Photo 1. I input the desired rotation and surface parameters and watch the anaglyph as it is drawn on the screen, executing Listing 3. The red wire-frame image is drawn first, followed by the blue one. If you look at a real-life three-dimensional surface with your left eye open and your right eye closed, then vice versa, you will see this image shift. You will see the same effect if you view the anaglyph through the filters and open your left eye and your right eye closed, then vice versa. Of course, in addition to

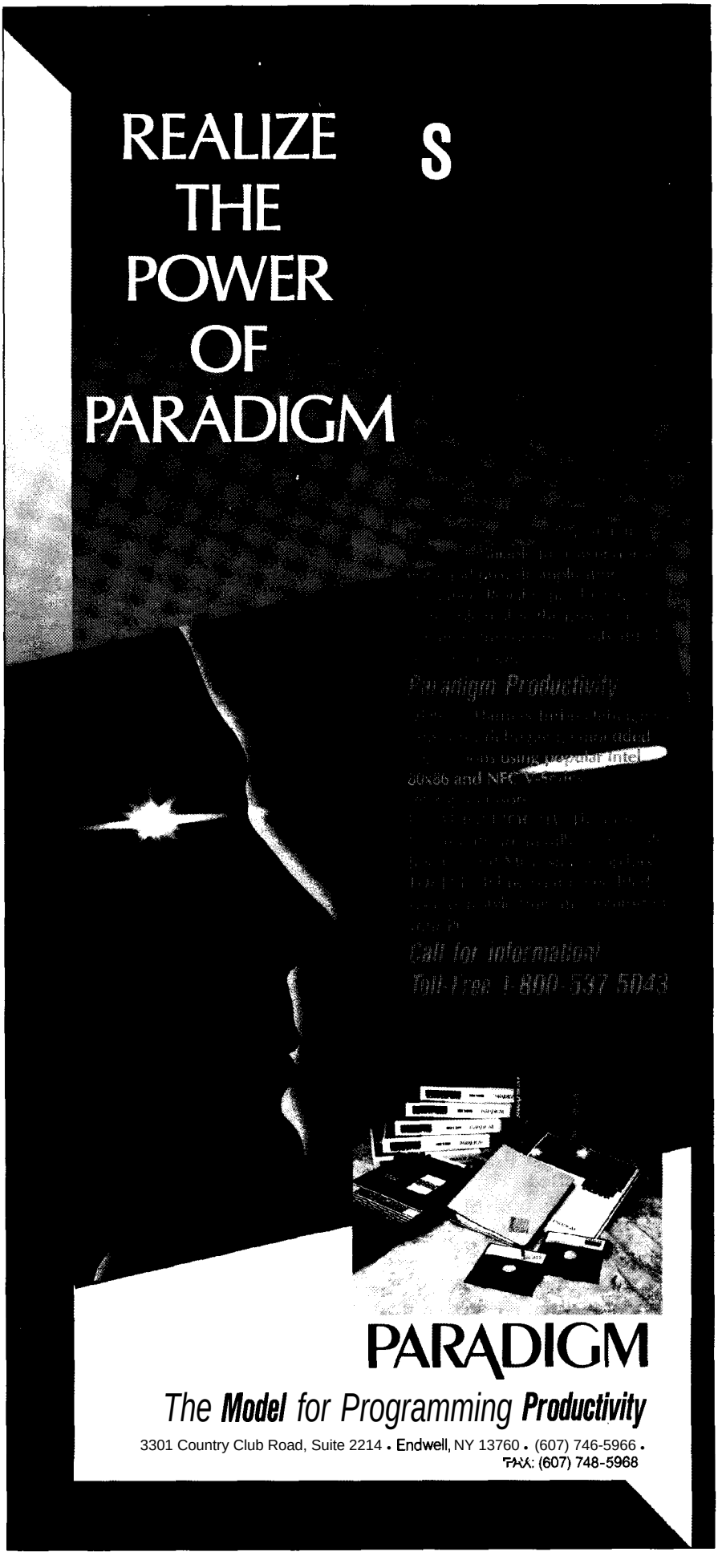
appearing repositioned for each eye, the images change colors.

Notice the drawings have many points where the blue intersects the red pels; these points should be transmitted to both eyes. Therefore, you must not overwrite the red pel with a blue one plotted later because critical right-eye information would be deleted. Simply plot a red and blue position with magenta (color red + blue). This way the pel information passes through both color filters and reaches both the left and right retinas.

While very few people have trouble seeing the three-dimensional effect, keep in mind a large part of the illusion depends upon your attitude. Although your brain knows there is no actual three-dimensional surface present, the anaglyphic coding usually overcomes this knowledge. You should see a magenta surface straddling the screen with no apparent relation to the glass surface when the two images are properly merged. Notice you focus and refocus your eyes when glancing at parts of the surface with different depths.

## REMOVING HIDDEN SURFACE POINTS

Notice in the anaglyph that the hidden lines, the ones that in the real world would be blocked from your view by the surface area between your eyes and the point in question, are all plotted. Unfortunately, the equation of the surface knows of nothing but the set of all points making up the surface. Creating an algorithm to perform this task is left to you, something that perhaps has the reputation of being "a royal pain," especially when mathematical surfaces are involved. I investigated several known methods and none of them appealed to me, so I set out to try it myself (with a mental note to prepare "to bail out and settle for what's available"). Although the analysis and solution of this problem consumed by far the greatest effort in this entire project, I got lucky and found a hidden line removal method that works for a mathematical surface of any rotation, executes fast, requires no buff-



ers, and is short and simple (the BASIC listing of the general procedure fits on a single screen).

A hidden-surface removal procedure can be either very simple or very complex, depending upon how you break down the problem and how each step of the procedure is solved. I've also found I have to try to form a mental block towards any known procedures when attempting to create a new one. This mindset helps promote design originality. Of course, you should use any known procedures that are satisfactory.

My procedure begins as follows: consider a surface point hidden if a line drawn from the point in question to the viewpoint intersects the surface elsewhere. Note in a wire-frame model a completely unplotted area can block the view of a point. In other words, treat the region bounded by a netted square as an opaque surface even though none of it is plotted. Thus, it appears as though you will be required to define every point, not just the netted regions, of the surface equation. But I found a slick way around this problem. As you plot the (rotated) surface, use only a subset of the entire surface in creating a wire-mesh model. Identify any hidden points by testing each point of the wire-frame model as it is drawn. The test is made by defining a space line with endpoints of the current surface point and the viewpoint. If you can move along the line from the viewpoint to the surface point without intersecting the surface elsewhere, then the point at the end of the line is visible; if you intersect the surface en route, then the point is hidden and should not be plotted.

Careful consideration of this procedure as applies to a computer program should make obvious a serious obstacle: the digital computer must find the precise intersection of a line with any part of a rotated surface. Therefore, every resolution element of the unbroken line must be tested. As a further complication, the surface has no particular relationship to the line points. Even if you test for values within a 0.001 unit error you would miss a surface-line intersection

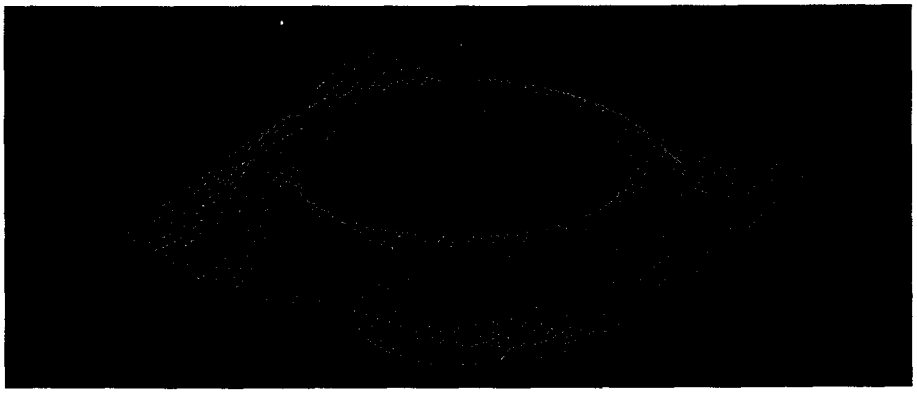


**Photo 3—*The display generated by the program in Listing 4.***

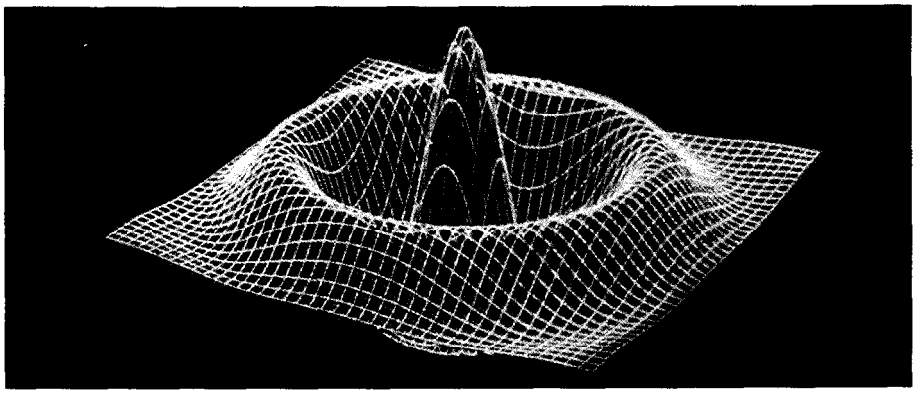


**Photo 4—*The finalanaglyph with hidden lines removed as generated by Listing 5 should be viewed with red/blue glasses.***

```
REM SURFACE (HIDDEN LINES REMOVED)--------1024x768
'$INCLUDE: 'gxlib.bas'
'$INCLUDE: 'grlib.bas'
retcode% = gxSetDisplay%(gxTS.38)
retcode% = gxSetMode%(gxTEXT)
PI = 3.1416: OOB = 0
INPUT "E,D,A,B,XI,XF,DX,YI,YF,DY "; E, D, A, B, XI, XF, DX, YI,
      YF, DY
INPUT "Amp, Dec, Freq "; TI, U, V
A = PI * A / 180: B = PI * B / 180
cosa = COS(A): cosb = COS(B) : sina = SIN(A): sinb = SIN(B)
retcode% = gxSetMode%(gxGRAPHICS)
retcode% = grSetViewPort%(0, 0, 1023, 767)
retcode% = qrSetViewWorld%(-511, -383, 511, 383)
retcode% = grSetWorld%(gxTRUE)
retcode% = grClearViewPort%
FOR mesh = 1 TO 2
FOR X = XI TO XF STEP DX: XSQ = X * X
FOR Y = YI TO YF STEP DY: YSQ = Y * Y

Z1 = TI * (EXP(-(U * ((XSQ) + (YSQ)))))
22 = COS(V * (SQR((XSQ) + (YSQ))))
Z = Z1 * 22

XAB = X * cosa * cosb - Y * sina * cosb t Z * sinb
YAB = X * sina + Y * cosa
ZAB = -X * cosa * sinb t Y * sina * sinb + Z * cosb

TL = D / (D - XAB)
YP = CINT(E + ((YAB - E) * TL))
ZP = CINT(ZAB * TL)

pcolor% = grGetPixel%(YP, ZP)
IF pcolor% = grRED THEN GOTO 7

dist = SQR(((XAB - D) * (XAB - D)) + ((YAB - El * (YAB - E)) +
       (ZAB * ZAB))
```

*(continued)*

**listing 4—*Removing hidden lines from the surface plot results in a more realistic-looking object. It also enhances the three-dimensionaleffect. The hidden-lineremovaltechniques used here are unique and work well.***

```
STP = -(30 / dist): RES = 1
TAL = 0

222 FOR t = 1 TO 0 STEP STP
TAL = TAL t 1
XL = D t ((XAB - D) * t)
YL = E + ((YAB - E) * t)
ZL = ZAB * t

sina = -sina: sinb = -sinb
XBA = XL * cosa * cosb - YL * sina + ZL * cosa * sinb
YBA = XL * sina * cosb + YL * cosa + ZL * sina * sinb
ZBA = -XL * sinb + ZL * cosb
sina = -sina: sinb = -sinb

ZUNA = TI * (EXP(-(U * ((XBA * XBA) + (YBA * YBA)))))
ZUNB = COS(V * (SQR((XBA * XBA) + (YBA * YBA))))
ZUN = ZUNA * ZUNB

DIFF = SGN(CINT(ZBA) - CINT(ZUN))
IF TAL < 3 THEN GOT0 22
IF TAL = 3 THEN ref = DIFF
IF DIFF - ref <> 0 THEN GOT0 7
IF ((XBA >= XF) OR (XBA <= XI) OR (YBA >= YF) OR (YBA <= YI))
   THEN OOB = 1
IF (OOB = 1) AND (RES = 1) THEN STP = STP / 64: TAL = 0: RES =
   0: GOT0 222
IF (OOB = 1) AND (RES = 0) THEN retcode% = grPutPixel%(YP, ZP,
   grRED): OOB = 0: EXIT FOR
22 NEXT t
7 NEXT Y, X
SWAP DX, DY
NEXT mesh
BEEP
DO: LOOP WHILE INKEY$ = ""
retcode% = gxSetMode%(gxTEXT)
END
```

**listing *4-continued***

```
REM ANAGLYPH (HIDDEN LINES REMOVED)--------1024x768
'$INCLUDE: 'gxlib.bas'
'SINCLUDE: 'grlib.bas'
retcode% = gxSetDisplay%(gxTS.38)
retcode% = gxSetMode%(gxTEXT)
PI = 3.1416: OOB = 0
INPUT "E,D,A,B,XI,XF,DX,YI,YF,DY "; E, D, A, B, XI, XF, DX. YI,
      YF, DY
INPUT "Amp,Dec,Freq "; TI, U, V
A = PI * A / 180: B = PI * B / 180
cosa = COS(A): cosb = COS(B): sina = SIN(A): sinb = SIN(B)
retcode% = gxSetMode%(gxGRAPHICS)
retcode% = grSetViewPort%(0, 0, 1023, 767)
retcode% = grSetViewWorld%(-511, -383, 511, 383)
retcode% = grSetWorld%(gxTRUE)
retcode% = grClearViewPort%

FOR image = 1 TO 2 ' two different perspectives
FOR mesh = 1 TO 2
FOR x = XI TO XF STEP DX: XSQ = x * x
FOR y = YI TO YF STEP DY: YSQ = y * y

IF image = 1 THEN COLOUR = 4 ELSE COLOUR = 1

I************ surface equation *********
21 = TI * (EXP(-(U * ((XSQ) + (YSQ)))))
22 = COS(V * (SQR((XSQ) + (YSQ))))
Z = Z1 * Z2

'************************ Alpha/Beta rotation equations ********
XAB = x * cosa * cosb - y * sina * cosb + Z * sinb
YAB = x * sina + y * cosa
ZAB = -x * cosa * sinb t y * sina * sinb + Z * cosb

I****************** screen projections************************
TL = D / (D - XAB)
YP = CINT(E t ((YAB - E) * TL))
ZP = CINT(ZAB * TL)
```

*(continued)*

**Listing 5—*Adding* the hidden-line removal routines *to* the *anaglyph* program *results* in *the final image.***

**occurring** digitally in the surface equation at 7.3614 and on the line equation at 7.3616. The omission would allow a hidden point to remain visible.

This problem exists because you must do exactly what you want to avoid: you must find exact intersections of the line with the surface. The answer is the reduction of the intersection-location requirement to determine whether a line point is either above or below the surface. This solution would be simple if the surface were unrotated, then you could just test if each line point is either greater than (above) or less than (below) the surface by comparing the line point with the surface equation evaluated at the same point. But you can't just test a point's position regarding a rotated surface because a vertical line may cut the surface many times. Any point of a vertical line intersects the surface at only one place in a true (unrotated) mathematical function; therefore, all other points of the line must lie above or below the surface, so you need only locate a sign change. How do you determine if a given point on the view line is located above or below the rotated surface? Just unrotate the surface point in question (remember, you define a particular surface point of the wire-frame model and test it), which you can operate as a true mathematical function. But be careful; you must also unrotate the point on the view line (effectively unrotating your viewpoint), so the entire surface and line perspective are preserved in the unrotated translation.

You can now perform an easy sign comparison for the entire view line or until the surface point is revealed as hidden, then rotate the point back and move on to the next one. Now you should know why you don't need to test any particular surface points; all you have to do is test the general surface equation for a signed magnitude relative to any viewpoint.

Next the question of how the surface (point) is unrotated arises. Can you simply swap the a and β parameters and reverse their signs? No! Order of rotation is significant (think

```
'************************* Don't waste tine replotting *********
PCOLOR% = grGetPixel%(YP, ZP)
IF image = 1 AND PCOLOR% = 4 THEN GOT0 7
IF image = 2 AND PCOLOR% = 1 THEN GOT0 7
IF image = 2 AND PCOLOR% = 4 THEN COLOUR = 5
IF image = 2 AND PCOLOR% = 5 THEN GOT0 7

dist = SQR(((XAB - D) * (XAB - D))+ ((YAB - E) * (YAB - E))+
        (ZAB * ZAB))
STP = -(30 / dist): RES = 1 'initally coarse step
TAL = 0
222 FOR t = 1 TO 0 STEP STP 'scan in surface-to-eye direction
TAL = TAL t 1

'*************************** 3D line equations *****************
xl = D t ((XAB - D) * t)
yl = E t ((YAB - E) * t)
zl = ZAB * t

'********************** unrotate: Beta/Alpha sequence **********
sina = -sina: sinb = -sinb
XBA = xl * cosa * cosb - yl * sina t zl * cosa * sinb
YBA = xl * sina * cosb t yl * cosa + zl * sina * sinb
ZBA = -xl * sinb + zl * cosb
sina = -sina: sinb = -sinb

'******************* unrotate surface point *********************
zuna = TI * (EXP(-(U * ((XBA * XBA) + (YBA * YBA)))))
zunb = COS(V * (SQR((XBA * XBA) t (YBA * YBA))))
zun = zuna * zunb

diff = SGN(CINT(ZBA) - CINT(zun)) 'reference line point to
surface
IF TAL < 3 THEN GOT0 22 'escape surface tension
IF TAL = 3 THEN ref = diff
IF diff - ref <> 0 THEN GOT0 7 'point is hidden

'*** set hi-res line scan and check for out of bounds (OOB) ****
IF ((XBA >= XF) OR (XBA <= XI) OR (YBA >= YF) OR (YBA <= YI))
   THEN OOB = 1
IF (OOB = 1) AND (RES = 1) THEN STP = STP / 64: TAL = 0: RES =
   0: GOT0 222
IF (OOB = 1) AND (RES = 0) THEN retcode% = grPutPixel%(YP, ZP,
   COLOUR): OOB = 0: EXIT FOR
'*****************************************************************
22 NEXT t

7 NEXT y, x
SWAP DX, DY
NEXT mesh
E = -E 'switch eyes
NEXT image
BEEP
DO: LOOP WHILE INKEY$ = ""
retcode% = gxSetMode%(gxTEXT)
END
```

listing **5**—*continued*

about it) and a $\beta/\alpha$ (tilt then twist) rotation is found in a manner similar to that for the original rotation. If you're following closely, you may be thinking for every surface point you have to do multiple rotations and test every point in a line for that plotted surface point--that's a lot of number crunching. Well, I lucked out here, too. Notice in the rotation formulas that a and $\beta$ are fixed from the start of the program, which means no trigonometric functions need be evaluated in the rotation formulas if you assign a and $\beta$ to fixed variables. This action results in a dramatic speed increase. As far as testing the line points, if you find a hidden line, then you need test that view line no more. To take advantage of this fact, you can originally step in large increments along the line while looking for a sign change. If you find the change, quickly move to the next viewpoint. If the coarse run shows no sign change, then you must fine test. This procedure also greatly improves drawing speed because a typical surface contains a large percentage of hidden points.

One other trick to speed things up: you never have to evaluate the entire length of the viewing line if, in the unrotated evaluation, you simply test the current point on the view line as being inside the domain $(xy)$ of the surface definition. This test will normally yield great speed improvement because viewing distance usually extends far beyond surface domain. These features can be seen in Listing 4, which draws a parallel perspective surface with the hidden lines removed. Watching this drawing form is interesting.

Listing 5 draws anaglyphic images with the hidden lines removed. Be sure you don't reverse the red and blue filters when viewing this one!

## THE FUTURE

It's surprising how simple the final code ends up being to generate threedimensional images on the computer screen. I have more ideas that I want to try along the same lines. I'll let you know if I come up with anything interesting.

## SOURCE

All the programs listed in this article are written in QuickBASIC with a third-party graphics library to achieve resolutions higher than the standard 640 x 480. For information on the GX Graphics library used with these programs, contact

Genus Microprogramming
11315 Meadow Lake
Houston, TX 77077
(800) 227-0918
(7 13) 870-0737
Fax: (7 13) 870-0288

*Dale Nassar has a B.S. in physicsfrom Southeastern Louisiana University (SLU). His hobbies includegymnastics (university team) and springboard diving.*

**IRS**

401 Very Useful
402 Moderately Useful
403 Not Useful

FEATURE ARTICLE

Chris Ciarcia

# PC-Based Optical System Design

## Using Your PC to Design a Lens

My normal 8-to-5 job requires that I work in an environment where there are over 7000 scientists and engineers, of which more than 3000 have Ph.D.s; and we all rub shoulders and work together in a small town on the side of a **mountain.** In such a 'brilliant" atmosphere one would expect to find a world of creativity, imagination, and constructive innovation. But in reality one often finds politics, professional suspicion, and an enormous amount of inertia. I suspect the first two are just manifestations of man's basic nature, but the presence of a great deal of resistance to progressive change makes me wonder why most of us spent ten years in graduate school "learning to think."

Of course it's not as bad as I imply. There are lots of scientists and engineers doing outstanding work in their fields. But from my experience within the field of computational science and engineering, most of the professionals I know wouldn't even consider starting a project without hundreds of thousands of dollars and unlimited access to some super mainframe computer. And if one is crazy enough to suggest that the PC on their desk (which is primarily used as a terminal and word processing system) might provide them with a powerful tool to complete a large portion of their task, you can expect to become very unpopular in the group office, especially while they're trying to justify their latest big budget.

There just seems to be this ever present inertia against reapplying old (tried and true, so there must be some thing better) techniques, using new instrumentation and using one's own mind to create "simple" straightforward models of complex systems. Perhaps this is because the latter requires a level of understanding and effort on the part of the individual not all that common today?

So with these thoughts in mind I take pen (keyboard) in hand and try to explain how you can use your desktop PC to model a real-world system, without needing to call on the resources of a mainframe. Much of the thought and inspiration that enabled the last generation of scientists and engineers to creatively "model" a problem on the restaurant napkin can be usefully employed by experimenters today. They can use the new slide-rule-the PC-to enable a whole new dimension of 2-D visualization and calculation.

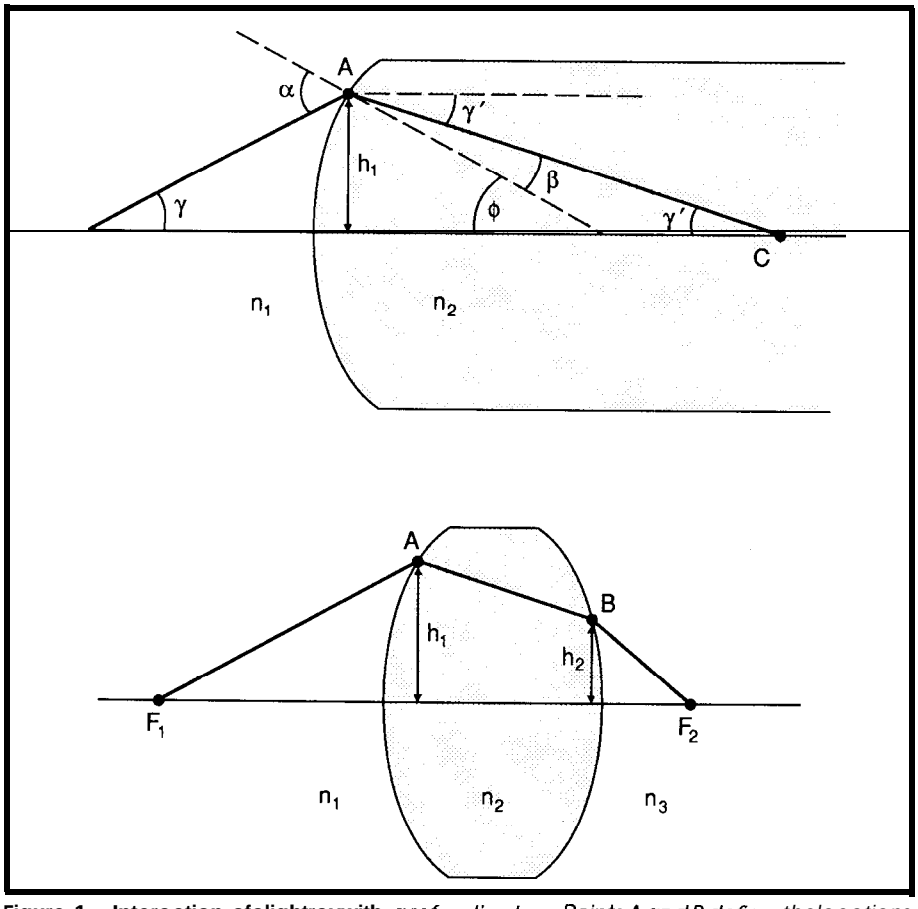



**Figure 1 --Interaction ofalightraywith** *a refracting lens. Points A and B define thelocations of the associated angles, heights, and focal points of the system.*

### DESIGNING A LENS

One of the primary components of most optical systems is the lens. Most of the time we employ off-the-

shelf optics to save on design and fabrication costs, but occasionally one needs to develop customized optics to get around some sort of design peculiarity or mechanical obstruction. In this case, laying out a simple geometrical design of the required lens becomes very useful. We need to choose basic parameters and specify requirements. Once this is accomplished, we can always go to the local optical fabrication shop to have our basic design optimized and built for us.

Doing this first "envelope-effects" calculation on our PC enables us to integrate our total design while we define operating limits. In spite of what you've heard, you don't need to use a mainframe to develop simple lens designs. Your PC is quite capable of doing matrix algebra, so it can be easily used for configuring, ray tracing, and estimating basic optical aberrations.

To start such a lens design, we must first examine the experimental environment and then determine the nature of the image light source, its optical frequency signature, the imaging plane, and the mechanical limitations inherent within the system. From this information we can make quick estimates of the necessary entrance and exit aperture sizes, optical path distances, the required focal lengths, the optical magnification (or reduction ratio), the F/number, the amount of allowed aberrations (input vs. output resolution), and so on. Once we have decided the basics, we need to create a simple model on the PC and trace light rays through the system to determine its overall abilities. This model should allow us to select specific light rays, add or remove elements, and change the element spacings or glass types (which affects index of refraction), so that we can modify and optimize to our particular requirements.

To do all of the above, we don't need to be optical engineers. We only need to understand a few optical basics, a little matrix algebra, and how to program our PC. Following such a matrix algebra procedure, we can create a simple program that will trace

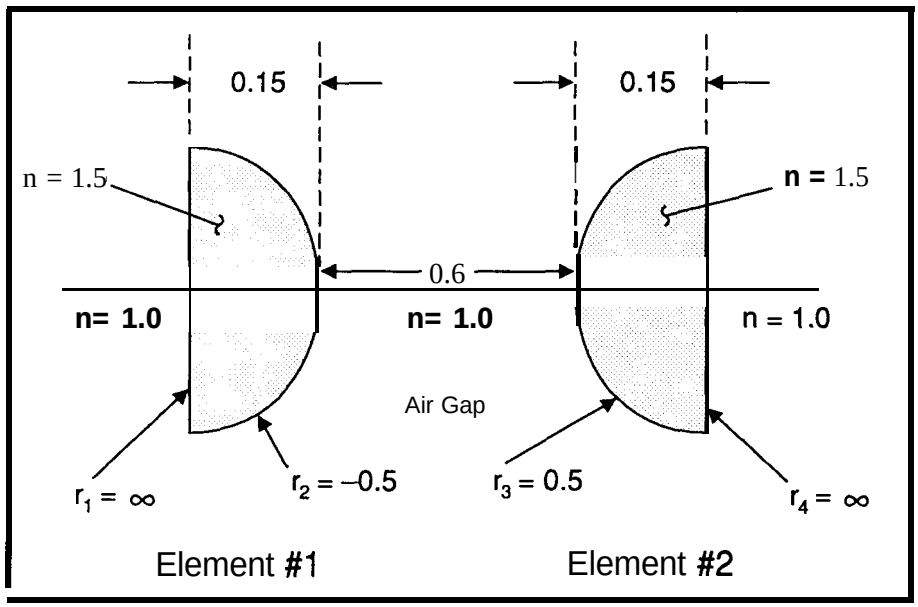



**Figure 2-A *two-element symmetrical lens system called the Ramsden eyepiece.***

the progress of a ray as it passes through one refracting surface after another in order to determine the height and direction it takes as it exits the last surface.

The implementation of such a design approach is based on the creation of two types of matrix operators, which represent the interaction of the light ray with the various interfaces within a lens. These are called the refraction (*R*) and translation (*T*) matrices. From these we can create a simple set of equations that are easily programmable, even for very complex compound lens systems.

## USING MATRIX ALGEBRA IN LENS DESIGN

To understand how to implement the technique I just described, consider the lens shown in Figure 1. Here we wish to follow a ray passing through the lens at points *A* and B. Just prior to entering the lens, the ray has an initial slope angle of g (as subtended with the optic axis). After refraction and while it is within the glass, it has a slope angle of $\gamma'$ with the ray intersecting the two surfaces at heights $h_1$ and $h_2$, respectively.

If we now define the angle f as the angle between our incident ray and the normal to the radius of curvature (*R*) of the lens at point *A*, then the angle $\phi$ can be defined as

$$\sin\phi = \frac{h_1}{R}$$

or in a first-order approximation

$$\phi = \frac{h_1}{R}$$

Now, according to Snell's Law and geometrical construction (see Figure la),

$$n_1 \sin a = n_2 \sin b$$

$$\alpha = \gamma + \phi$$
$$\beta = \gamma' + f$$

then Snell's Law can be rewritten in the form

$$n_1\left(\gamma + \frac{h_1}{R}\right) = n_2\left(\gamma' + \frac{h_1}{R}\right)$$

If we now solve the above equation for $n_2\gamma'$, then

$$n_2\gamma' = n_1\gamma + \frac{n_1h_1}{R} - \frac{n_2h_1}{R}$$
$$= n_1\gamma + \frac{(n_1 - n_2)h_1}{R}$$

and since $(n_2 - n_1)/R$ is the refractive power *P*,

$$n_2\gamma' = n_1\gamma - Ph_1 \qquad (1)$$

and using the identity

$$h_2 = h_1 \qquad (2)$$

REALIZE THE POWER OF...




Call for information!
Toll-Free 1-800-537-5043
PARADIGM
The Model for Programming Productivity
3301 Country Club Road
Suite 2214
Endwell, NY 13760
(607) 748-5966
FAX: (607) 748-5968
All trademarks are property of their respective holders.

we can construct the refraction matrix by writing Equations 1 and 2 in matrix form:

$$\begin{vmatrix} 1 & -P \\ 0 & 1 \end{vmatrix} \begin{vmatrix} n_1\gamma \\ h_1 \end{vmatrix} = \begin{vmatrix} n_2\gamma \\ h_1 \end{vmatrix}$$

where, as a 2 x 2 matrix, it can be written as

$$\begin{vmatrix} 1 & -P \\ 0 & 1 \end{vmatrix} = R$$

All ray interactions with the surface of a lens element can now be characterized using this refraction matrix. But we must now consider the motion of the ray as it passes through the glass (or an air gap) going from point *A* to *B*. If *d* is the horizontal distance between *A* and *B* then,

$$h_2 - h_1 = d \tan \gamma'$$

and for paraxial rays (a ray trace based on the paraxial approximation in which all angles are assumed equal to their sines and tangents) this equation becomes,

$$h_2 = h_1 + d\gamma' \qquad (3)$$

At *A*, the ray emerges in a direction $n_2\gamma'$ with height $h_1$ above the optic axis; while at *B* the angle of incidence is equal to the emergent angle from *A*. Realizing that $\gamma'$ is the same at *A* and *B*, then we can again write an identity equation like that in Equation 2 where,

$$n_2\gamma' = n_2\gamma' \qquad (4)$$

now, writing these two equations in a simple matrix form,

$$\begin{vmatrix} 1 & 0 \\ d/n_2 & 1 \end{vmatrix} \begin{vmatrix} n_2\gamma \\ h_1 \end{vmatrix} = \begin{vmatrix} n_2\gamma \\ h_2 \end{vmatrix}$$

which becomes our translation matrix *T*:

$$\begin{vmatrix} 1 & 0 \\ d/m & 1 \end{vmatrix} = T$$

If we now follow our selected ray through the lens in a step-by-step fashion, we can create a composite matrix that incorporates the total action of the lens. For this example, the system matrix is a combination of the two refraction matrices resulting from ray interaction with the front and rear surfaces of the lens and the one translation matrix as it passed through. This is defined as the matrix product of each operation matrix ordered from right to left,

$$S = R_2 T R_1 \qquad (5)$$

If there is more than one lens, then these matrices are cascaded to-

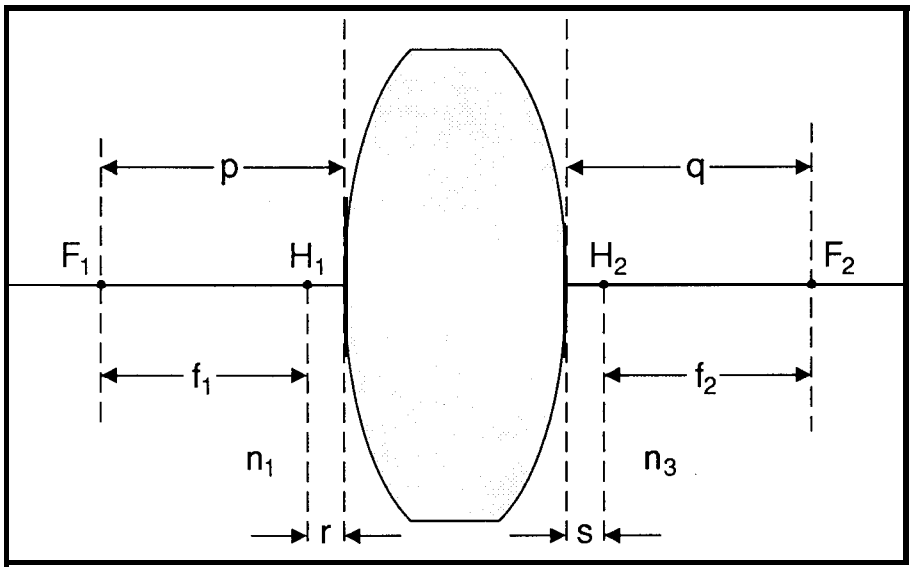



**Figure 3**—*Definition of the principal planes (H) and the focal planes (F) of an optical system.*

```
$debug

c       A,B,C,T,SE are 2x2 matrices used for temporary storage
c        of the translation and refraction matrices of the lens,
c        S = system matrix
c       surf,radius,thk,index are the lens design date from the
c        input file
c       p,q,r,sz,f1,f2,h1,h2 are geometrical parameters
c       a_in = gamma, a_out=gamma'
c       ne = number of elements and nk=number of records in the
c        design file
c       lname is the lens design filename

        real*4          A(2,2),B(2,2),C(2,2),T(2,2),S(2,2),SE(2,2)
        real*4          surf(100),radius(100),thk(100),index(100)
        real*4          p,q,r,sz,f1,f2,h1,h2,a_in,a_out
        integer*2   ne,nk
        character*22   lname

        write(*,1)
1       format(1x,'Input lens data filename: ',\)
        read(*,'(a22)')lname
        write(*,5)
5       format(1x,'Input incident ray height and slope angle.',/,
     *          1x,'h1,gamma: ',\)
        read(*,*)h1,a_in

        open(1,file=lname,status='old')
        k=1
10      read(1,*,err=20)surf(k),radius(k),thk(k),index(k)
        k=k+1
        goto 10
20       close(1,status='keep')

        k=k-1                       ! decrement record counter
        nk=k                        ! save number of records
        ne=k/3                      ! number of elements
        k=k-1                       ! set to last element surface

        print  *,''                 ! R = refraction matrix
                                    ! T = translation matrix
        do j=1,ne

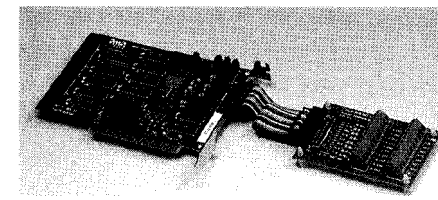
*RT      ************************************************************
        A(1,1) = 1
        A(2,2) = 1
        A(2,1) = 0
        if(radius(k).eq.0)then
           A(1,2)=0
        else
           A(1,2) =  (index(k)-1.0)/radius(k)
        endif
        B(1,1) = 1
        B(2,2) = 1
        B(1,2) = 0
        B(2,1) = thk(k)/index(k)
        call mult(A,B,C)
* RTR    ************************************************************
        A(1,1) = 1
        A(2,2) = 1
        A(2,1) = 0
        if(radius(k-1).eq.0)then
           A(1,2)=0
        else
           A(1,2) = - (index(k)-l.0)/radius(k-1)
        endif
        call mult(C,A,B)                ! RTR
* RTRTa ************************************************************
        A(1,1) = 1
        A(2,2) = 1
        A(1,2) = 0
        A(2,1) = thk(k-2)/index(k-2)
        if(ne.eq.1)then                 ! if only 1 element
            call mult(B,A,s)
        endif
        if(j.eq.1)then                  ! if more than 1 element
            call mult(B,A,se)
        else
           call mult(B,A,t)
        endif
*****************************************************************
        k=k-3
        if(j.gt.1)then
           call mult(se,t,s)
           do m=1,2
```

(continued)

**Listing 1** -*A FORTRAN program for simple lens design*


# Data Acquisition Solutions

**Reliable, Affordable and Flexible**



**PCL-711S 12 Bit A/D + D/A $2951**

- 8 single-ended 12-bit analog inputs, 2.5 KHz
- 1 D/A, 16 digital inputs, 16 digital outputs
- Wiring terminal board and cable included

**PCL-812 A/D + D/A + DIO $395**

- 16 single-ended 12-bit analog inputs, 30 KHz
- A/D with multiple ranges, DMA, interrupt
- 2 D/A, 16 D/I, 16 D/O, counter, pacer
- QBASIC driver, utility software and manual

**PCL-818 High Speed A/D $875**

- 16 S.E. or 8 differential 12-bit A/D, 100 KHz
- Programmable ranges, auto channel scanning
- A/D with direct I/O, interrupt or DMA
- 2 D/A, 16 D/I, 16 D/O, counter, pacer
- Supported by most popular DAS software

**PCL-726 6 Ch. 12-Bit D/A $495**

- 6 12-bit analog outputs, voltage or 4 - 20 mA
- 16 digital inputs and 16 digital outputs, utility

**PCG722 144 Ch. Digital I/O $345**

- 144-channel digital I/O with interrupt
- Emulating 82.55 mode 0, buffered circuits for high driving capacity, I/O dir. programmable
- Compatible with OPTO-22 I/O racks

**PCG720 DIO and Counter $175**

- 32 D/I, 32 D/O, 3 programmer counters
- User selectable counter clock source
- Breadboard area for customized circuits

**Signal Conditioning and more . . .**

- Wiring terminal, isolated D/I, relay output, relay driver, scanner, multiplexer, amplifier
- Interface for thermocouple, RTD, strain gage
- IEEE-488, RS-232/422/485, stepping motor
- Industrial PC chassis, 286/386/486 CPU cards, RAM/ROM disks, PC-bus card cage




**Free 120-page reference guide for your system and OEM needs**

Industrial & Lab Automation with PCs

**ADVANTECH**

1310 Tully Rd., Suite 115
San Jose, CA 95122, USA
(408) 293-6786, FAX (408) 293-4697

Reader Service #105

gether and multiplied out to create the total system matrix for a set of individual elements within a compound lens.

Multiplication of these 2 x 2 matrices is very simple and easy to code, as shown below. Note that the product of any number of 2 x 2 matrices is by itself a 2 x 2 matrix.

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} \begin{vmatrix} e & f \\ g & h \end{vmatrix} = \begin{vmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{vmatrix}$$

The path of a ray can be determined at each interface, or at the exit surface of the last element by applying the system matrix in the following manner:

$$S \begin{vmatrix} n_1\gamma \\ h_1 \end{vmatrix} = \begin{vmatrix} n_n\gamma' \\ h_n \end{vmatrix}$$

where $h_1$ and $\gamma$ are the height and slope angle of the incident angle striking the first element in a compound lens, and $h_n$ and $\gamma'$ are the height and slope angle of the ray exiting the last element.

To demonstrate how simple this matrix technique is to apply, consider the example shown in Figure 2, called a Ramsden eyepiece. If we apply our matrix forms defined above, the system for this compound lens becomes

$$R_4T_2R_3T_aR_2T_1R_1 = S \qquad (6)$$

here, $T_a$ is the translation matrix for the air gap between elements 1 and 2. Solving for the refractive power for each surface and substituting these into Equation 6, the system matrix for this two-element lens becomes,

$$S = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 \\ 0.1 & 1 \end{vmatrix} \begin{vmatrix} 1 & -1 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 \\ 0.6 & 1 \end{vmatrix} \begin{vmatrix} 1 & -1 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$$
$$= \begin{vmatrix} 0.260 & -1.400 \\ 0.666 & 0.260 \end{vmatrix}$$

But this system matrix can be used for other purposes than just finding the path of a ray through a compound lens. It can be used to find much of the geometry of the system (e.g., the focal planes [F] and the principal planes [H]).

The four elements of a system matrix compose what is known as the Gaussian constants--u, b, c, d-of a system. They are arranged as

$$S = \begin{vmatrix} b & -a \\ -d & c \end{vmatrix}$$

These constants relate to the above mentioned physical dimensions (see Figure 3). The distances between the two focal planes, *F*, and the respective lens surfaces are

$$p = \frac{-bn_1}{a} \quad b \quad f$$

$$q = \frac{cn_3}{a}$$

```
            do n=1,2
               se(m,n)=s(m,n)
            enddo
            enddo
         endif
         enddo

******* geometry ****************************************

         r  =  index(1)*(  (1-s(1,1))/(-1.0*s(1,2)) )
         sz =  index(nk)*(  (s(2,2)-1)/(-1.0*s(1,2)) )
         p =  -1.0*s(1,1)*index(nk)/(-1.0*s(1,2))
         q =       s(2,2)*index(nk)/(-1.0*s(1,2))
         f1 = p-r
         f2 = q-sz

         a_out = s(1,1)*index(1)*a_in + s(1,2)*h1
         a_out =  a_out/index(nk)

         h2    = s(2,1)*index(1)*a_in + s(2,2)*h1

******* output ******************************************

         print  *,''
        call out(s)

         print *, 'System Geometry'
         print *,'--------------------------'
         print  *,' r= ',r
         print  *,' s= ',sz
         print *,' p= ',p
         print  *,'q= ',q
         print *,'f1= ',f1
         print *,'f2= ',f2
         print *,'----------------------------------'
         print *,'ray incident  angle  = ',a_in
         print *,'ray exit      angle  = ',a_out
         print *,'ray incident height = ',h1
         print *,'ray exit      height = ',h2
         print *,'----------------------------------'
         end

         subroutine mult(m1,m2,m3)
c        this subroutine multiplies 2 matrices m1xm2
c        output of operation = m3
         real*4 m1(2,2),m2(2,2),m3(2,2)
         m3(1,1)=m1(1,1)*m2(1,1) + m1(1,2)*m2(2,1)
         m3(2,1)=m1(2,1)*m2(1,1) + m1(2,2)*m2(2,1)
         m3(1,2)=m1(1,1)*m2(1,2) + m1(1,2)*m2(2,2)
         m3(2,2)=m1(2,1)*m2(1,2) + m1(2,2)*m2(2,2)
         return
         end

         subroutine out(m)
c        this subroutine prints out the system matrix
         real*4 m(2,2)
            print *, 'System Matrix (Gaussian constants)'
            print  *,''
            print   *,' b               -a'
            print *,'----------------------------------'
            do i=1,2
                print *,(m(i,j),j=1,2)
            enddo
            print * ,'----------------------------------'
            print   *,' -d               c'
            print   *,''
         return
         end
```

**listing** 1 —*continued*

The distances between the principal planes, $H$, and the lens vertices are

$$r = n_1\left(\frac{1-b}{a}\right)$$

$$s = n_3\left(\frac{c-1}{a}\right)$$

The difference of these two distances on either side, $p-r$ and $q-s$, then gives the focal lengths.

To demonstrate, consider our example lens in Figure 2. To determine the focal length of this compound lens, we start by locating the principal planes of the "total system,"

$$r = n_1\left(\frac{1-b}{a}\right) = 1.0\left(\frac{1-0.260}{1.40}\right) = +0.529$$

$$s = n_3\left(\frac{c-1}{a}\right) = 1.0\left(\frac{0.260-1}{1.40}\right) = -0.529$$

furthermore,

$$p = \frac{-bn_1}{a} = \frac{-0.260}{1.40} = -0.186$$

$$q = \frac{cn_3}{a} = \frac{0.260}{1.40} = +0.186$$

and the two focal lengths are easily found as

$$f_1 = p - r = (-0.186) - 0.529 = -0.715$$
$$f_2 = q - s = 0.186 - (-0.529) = +0.715$$

with the units given in whatever system used throughout.

Finding the path of any ray through the lens is equally easy. If we assume that a ray is incident at height $h_1$ and slope angle $\gamma$, this ray exits the lens at slope angle

$$\gamma' = 0.260\gamma - 1.4h_1$$

and the exit height is

$$h_2 = 0.666\gamma + 0.260h_1$$

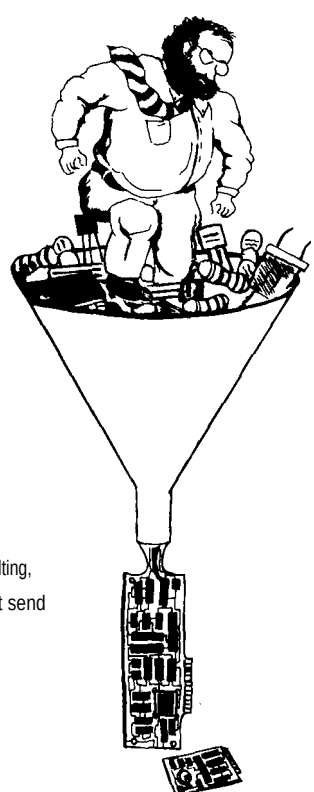
## SAMPLE LENS DESIGN CODE

To demonstrate how easy it is for you to do simple lens design on your PC, consider the program (MATRIX) shown in Listing 1. This program is designed to read in a simple lens data file; find the refraction and translation matrices for each surface, element, and air gap within a compound lens; find the principal planes and focal planes of the optical system; and then the exit height and direction of an incident ray. In its current form it can be easily incorporated into a graphics program which might be designed to draw each surface and graph the path of a ray (or rays) as it (or they) transverse the lens. This would enable you to define the appropriate apertures (F/number) and stops. Its fundamental usefulness lies in its ability (even in its present form) to define the EFL (effective focal length, or the distance from the second or first principal point to the second or first focal point of the optical system) and thereby the system's power (1/EFL), as well as the other geometrical design parameters needed for optical fabrication (without needing a megabuck design program like CODE-V).

| surface | radius | thickness | index | |
|---|---|---|---|---|
| 0 | 0.00 | 0.000 | 1.00 | .. entrance |
| 1 | 0.00 | 0.000 | 1.00 | .. 1st surf, element 1 |
| 2 | -0.5 | 0.150 | 1.50 | .. 2nd surf, element 1 |
| 0 | 0.00 | 0.600 | 1.00 | .. air gap |
| 3 | 0.50 | 0.000 | 1.00 | .. 1st surf, element 2 |
| 4 | 0.00 | 0.150 | 1.50 | .. 2nd surf, element 2 |
| | | | | .. air gap |
| | | | | .. 1st surf, element n |
| | | | | .. 2nd surf, element n |
| 0 | 0.00 | 0.000 | 1.00 | .. exit |

Figure 4-A sample of on input file used by MATRIX for the Ramsden eyepiece shown in Figure 2.


# The Ciarcia Design Works



Does your big-company marketing department come up with more ideas than the engineering department can cope with? Are you a small company that can't afford a full-time engineering staff for once-in-a-while designs?

Steve Ciarcia and the Ciarcia Design Works staff may have the solution for you.

We have a team of accomplished programmers and engineers ready to design products or solve tricky engineering problems. Need an infrared remote controller, multi-channel data logger, or 7-axis drill controller? The team has the talent to design and manufacture it!

Whether you need an on-line solution for a unique problem, a product for a startup venture, or just plain experienced consulting, the Ciarcia Design Works stands ready to work with you. Just send me a fax discussing your problem and we'll be in touch.

**Remember...*a Ciarcia design works!***

**Call (203) 875-2199 . Fa (203)** 872-2204

```
Input lens data filename: lens.dat
Input incident ray height and slope angle.
h1,gamma: 0.3,22.7

System Matrix (Gaussian constants)
    b              -a
  ------------------------------
    2.600000E-01       -1.400000
    6.6599993-01    2.600000E-01
  --------------------------------
    -d              c

 System  Geometry
 ------------------------

  r=    5.2857153-01
  s=   -5.2857143-01
  p=   -1.8571433-01
  q=    1.8571433-01
 f1=   -7.1428573-01
 f2=    7.142857E-01
                        ------------------------
 ray  incident  angle  =       22.700000
 ray  exit      angle  =        5.481999
 ray  incident  height =    3.000000E-01
 ray  exit      height =       15.196200
                        ------------------------
```

**Figure 5**—*The output generated by* **MATRIX** when *the data is Figure* 4 is presented *to the* program.

A sample of an input file (LENS.DAT) used by MATRIX for the Ramsden eyepiece example described above is shown in Figure 4. When MATRIX reads the data file, it determines the number of elements within the system and then computes the system matrix S. It then computes some of the system geometrical parameters as well as the exit height and slope angle of an input ray. Figure 5 is an example of a MATRIX run for the Ramsden eyepiece example.

## ONWARD AND UPWARD

Similar techniques can be used to simulate other portions of a more complex optical system. Taken in combination, the performance of the entire system can be simulated before any part of it is ever fabricated without resorting to complicated Monte Carlo calculations on mainframe computers. Don't underestimate the power of your PC. It may be slower than a Cray, but it has more flexibility (and a lower price tag). I know: I work on Crays 20 to 30 hours a week. Besides, I like playing with my PC.

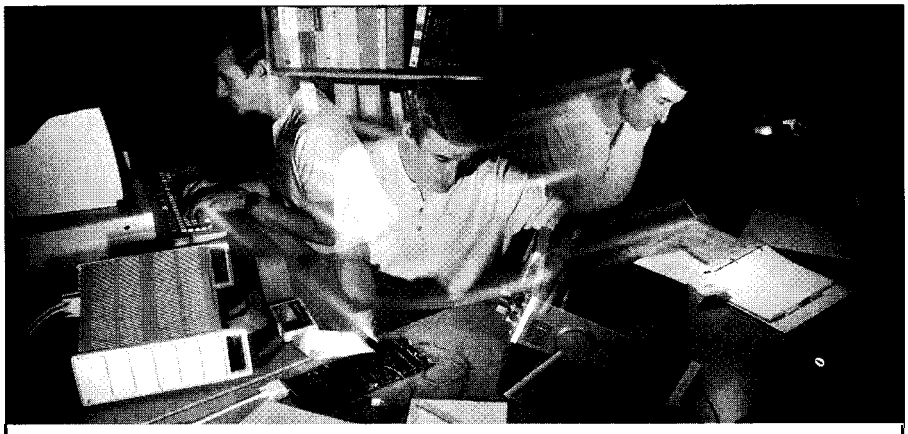
## REFERENCES


1. W. Brouwer. 'Matrix Methods in Optical Instrument Design," W.A. Benjamin, Inc., New York, 1964.
2. D.P. Feder, "Automatic Optical Design," Applied Optics 2 (1963). pp. 1209-26.
3. W.J. Smith, 'Modern Optical Engineering," McGraw-Hill, Inc. 1966, p. 308.



*Chris Ciarcia has a Ph.D. in experimental physics and is currently working as a* staff *physicist at a national lab. He has extensive experience in computer modeling* of *experimental systems, image processing, and artificial intellignece. Chris is also a principal in Tardis Systems.*





MultiTask!™ Executives
Accelerate Real-Time Design

In a hurry to develop real-time applications? MultiTask! executives give you a full set of standard system services for most embedded processors.

Source code keeps you in control. And our ProtoTask!™ executive lets you develop code on the PC. No matter which target you choose or when you choose it.

MultiTask! executives support today's most popular microprocessors:

680x0, 68HC11, Z80/Z180/64180, 8051, 80x86/V-Series, 8096/80196 & i960.

Call for a free EasyTask!™ information diskette: (800) 356-7097 or (503) 641-8446.

14215 NW Science Park Drive
Portland, OR 97229
U S SOFTWARE®

© 1991 US Software Corporation. MultiTask!, ProtoTask! and EasyTask! are trademarks of US Software Corporation.

Reader Service #20


## IRS

404 Very Useful
405 Moderately Useful
406 Not Useful

FEATURE ARTICLE Part 2

# A Video Editing System

William J. Kressbach

"Standby-quiet on the set—this'll be a take."

"Ready announce mic-ready to cue announce-ready to fade in on camera one...."

"Mic--cue-up on one."

In part one, I talked about how video tape is edited, the control mechanisms that make precision editing possible, building a Z8-based editing controller, and the controller's software. In this part, I'll discuss the software for the host computer and describe some rather unconventional ways to make editing easy and convenient.

"Ready talent n-tic-ready cue talent-ready to dissolve to two."

So here'sssssssssssss Bill.

"Mic-cue-dissolve to two.. .. Kill announce mic."

Spend some time in the video business and you'll discover the quality of video equipment is directly related to the number of zeros on the end of the price tag. My objective has been to build a project that will help reduce the number of zeros, while developing quality editing equipment that's functional and useful. Last time, I showed how to construct a pair of Z8-based controller modules, one for each of the two editing VCRs. The controllers are daisy chained through a serial I/O to a host computer. The host displays relevant information from each controller, and it translates and relays editing instructions to the controllers.

The software for the host computer is written in Z80 machine code and runs on the Radio Shack Model IV. I'll limit my discussion of software to general terms because you probably will want to write your own or translate to another system. While I have long since changed over to MS-DOS for most everything, I still do video editing on the trusty (I didn't say rusty) Model IV. The Model IV is no longer in wide distribution. If you don't own one, and you're not into translating machine code, consider looking around for a used one. Thousands of them must be sitting in closets unused and they can sometimes be found reasonably priced at garage sales. If you're an MS-DOS or CP/M user, you won't be uncomfortable with the operating system long. Also, there's no need for a hard disk, dual floppies will work just fine. You can read and write the disks on an IBM PC with a program called TRS-Cross. *[Editor's Note: Software for this article is available on the Circuit Cellar BBS and on Software On Disk #24. See page 105 for downloading and ordering information.]*

***Spend some time in the video business and you'll discover the quality of video equipment is directly related to the number of zeros on the end of the price***

## ROLLING YOUR OWN

Think about "rolling your own" when it comes to editing control functions. My methods of editing may be considerably different from yours. After all, I free-lance: I write the scripts, run the camera, log the work tapes, and do all the final editing. Your work may not be as extensive, but as long as you're going to build your own system, make it one to grow into in case you decide to expand later. I'll explain what I dreamed up and maybe you can improve on it.

The Z80 machine code is divided into eight different modules, each reflecting a different function. MS-DOS users who are unfamiliar with the Model IV will be happy to hear it's interrupt driven and uses DOS calls (called SVCs) in very much the same

| | IN | OUT | DURATION |
|---|---|---|---|
| STOP | 03:02:16-07 | 03:02:18-01 | 03:02:19-03 | 00:00:01-02 |
| PLAY | 01:02:04-05 | 01:02:10-24 | 01:02:11-26- | 00:00:01-02 |

Figure 1 --Editing information is arranged in left to right, top to bottom order. Top is the source VCR, bottom is the destination. Starting at the left is the VCR mode, current tape time, beginningeditpoint, ending edit point and edit duration. Thesymbols, PSRI for pause, slow, record, andinsert (not shown), are displayed just to the right of the current tape time. The operator may choose either the source or destination VCR for selection of the cut-out point b Y pressing the hyphen (-) key from the keyboard. The length of the edit for the other is calculated to match. The hyphen to the right of the destination cut-out time indicates which is the entered time code.

fashion as MS-DOS. These aspects, combined with similar processor architecture, make any translating fairly easy.

Before we dig into the software, let's examine what it does. I'll divide the editing process into three basic areas and describe how each is accomplished. These areas are: editing and VCR control, recording and handling of editing information, and logging work tapes.

## BASIC EDITING FUNCTIONS

Basic editing and status information is displayed across the bottom two lines of the screen, which leaves plenty of room on the rest of the screen for other uses. The editing portion of the screen display is shown in Figure 1. The layout is in a left-to-right, top-to-bottom configuration. Apply the same layout to the physical location of the editing machines and monitors as well; a good logical layout is the best way to avoid confusion.

Figure 2 lists the keys I've assigned for different control operations. Basic machine control functions, such as play, rewind, and fast forward, use the numerical keypad. The "stop" function is assigned to the <5> key, roughly approximating the relative location of the <stop> key on the VCRs I use. The <2> key is used to put the VCR into the play mode, which seemed logical because it is directly below the <stop> key. Fast forward is to the right of play, and rewind is to the left. Pause and slow are to the right and left of stop. Tapping the space bar is a "kill" command, used to stop both machines and cancel any commands currently in process.

The top row of number keys are assigned to the record functions. The <7> key is used for record, while the <8> and the <9> keys are used for cut-in and cut-out, respectively. When I replaced my original VCRs, the new ones had a reverse feature the old ones lacked. No more pins were left on the controller, so I reassigned the pin used for record over to reverse. If you plan on using the record func-

| | |
|---|---|
| / | Enters word processing mode-BREAK exits |
| ^D | Deletes letter if over time code marker deletes time code |
| ^C | Delete to end of file |
| ^E | Produce edit decision list |
| L | Read disk file |
| W | Write disk file |
| Q | Quit |
| n | Trim up--Increments destination cut-in time |
| N | Trim up-Increments source cut-in time |
| ^N | Trim up-Increments tape time under cursor |
| j | Trim down-Decrements destination cut-in time |
| J | Trim down-Decrements source cut-in time |
| ^J | Trim down-Decrements tape time under cursor |
| | Manual enter-destination cut-in |
| < | Manual enter-source cut-in |
| ^ | Manual enter-tape time under cursor |
| | Manual enter-destination cut-out |
| > | Manual enter-source cut-out |
| P | Move script to agree with tape |
| ^P | Lock script to tape and follow |
| f | Find and stop destination |
| F | Find and stop source |
| ^F | Find string in script |
| t | Enter destination tape time |
| T | Enter source tape time |
| x | Index tape (queries for index start time) |
| ^X | Index tape from current tape time |
| → | Moves cursor right |
| ^→ | Moves to next edit point in script |
| ← | Moves cursor left |
| ^← | Moves to last edit point in script |
| ^L | Locks edit points in script to current edit points |
| ^U | Unlocks edit points in script |
| ^K | Auto edit-(automatically edits list from cursor to EOF) |
| – | Toggles source of edit duration |
| . | Toggles between source = camera, source = tape |
| | Preview insert edit |
| <clear> i | Execute insert edit |
| a | Preview assemble edit |
| <clear> a | Execute assemble edit |
| r | Review edit |
| ^F1 | Move to next script marker (next time code in script) |
| F2 | Enter destination tape time into cut-in |
| Shift F2 | Enter source tape time into cut-in |
| ^F2 | Enter source tape time into script |
| F3 | Enter destination tape time into cut-out |
| Shift F3 | Enter source tape time into cut-out |
| ^F3 | Enter destination tape time into script |

Figure 2—*The* **controller supports** *a large set of editing control codes. Note the difference between upper and lower case in many of the commands. A carat preceding a command indicates a control character. A <clear> preceding a command indicates holding down the clear key. The <clear> key on the Model IV is often used in a similar fashion to the <Alt> key on an IBM PC.*

tion or doing any assemble editing, you'll have to find a new pin and key assignment.

The Model IV has three function keys placed just above the numerical keypad. I use these keys for quick entry of cut-in and cut-out locations from off tape. To use this function, put the destination VCR in play and watch for the place where you want the edit to start. When you see it, hit the <F2> key. The current tape time is saved in memory and displayed as the cut-in location on the display. When you come to the location on the tape that you want the edit to stop, press <F3>. The current tape time is then transferred to the cut-out display. Use the shifted <F2> and <F3> keys to mark locations from the source tape.

You can trim the cut-in and cut-out locations up or down, or enter edit numbers directly from the keyboard. Keep in mind that the key assignment should continue the top-to-bottom orientation as much as possible. Pressing the unshifted key always affects the bottom, or destination machine, while pressing the shifted key activates the top, or source machine. Holding down <Ctrl> and pressing a key sends the same command to both VCRs.

## KEEPING GOOD NOTES

Keeping good notes on your edit points is important because videos are frequently changed. One day, while I was writing edit points on a copy of a script, it occurred to me that I wrote my scripts on a computer and did my editing on a computer, but I was still using an old fashioned ballpoint pen to record my edit points. This thought led me to the rather unconventional system of inserting and extracting editing information directly to and from the script.

The best way to illustrate this system is through example. When I write a script I have a pretty good idea what it's going to look like and where edits are likely to occur. After the narrator records the script, I convert the script file to ASCII format and load it into the Model IV. Once the tape has been time coded, I go through the script and insert markers at the locations where edits are likely to occur. Simply move the cursor to the edit location and press <control +>. A <+> is automatically inserted into the script with extra spacing to help make it stand out.

Next, I go through the tape listening to the narration, exchanging each "+" for the corresponding time code from off tape. I've built in some code to make this exchange much easier to do than it sounds. Again the three function keys come into play. <Ctrl F1> is programmed to do a "find next +" function. <Ctrl F3> inserts a time code from the destination VCR into the script. If a "+" is under the cursor at the time, it is removed and replaced with a time code. The time code is enclosed by parentheses to distinguish it from the rest of the script.

You'll discover you can quickly go through a tape using the <F1> key to find markers and <F3> to insert time codes. A source VCR time code may also be inserted by pressing <Ctrl F2>, and it is bracketed by left and right carats. The software interprets a cursor over the left parenthesis of an existing destination time code to mean that video or audio from the source machine is to be inserted there. Therefore, the software assumes the two time codes belong together. The resulting time code pair is formatted in left-to-right orientation and will be interpreted as a pair in future operations. Figure 3 is a script segment showing inserted time codes.

Following this notation takes a bit of getting used to. If you have trouble, press <Ctrl E> to extract the time code pairs from the script and produce a more conventional Edit Decision List (EDL). The EDL is then appended to the end of the script and can be viewed by simply scrolling down. The code for creating the EDL is in the module NONAME. The module can be easily modified to emulate whatever EDL format you're accustomed to using. Figure 4 shows a typical EDL generated by the program.

Sometimes when marking time codes, you have to mark the location several times before you get it just right. The software is smart enough to recognize when a time code already exists

```
    <06:03:58-15> (01:01:35-10) FIRST, THIS MACHINE HAS TWO-HAND BUTTON
CONTROLS. IT <06:02:34-17> (01:01:38-26) REQUIRES TWO BUTTONS TO BE PRESSED
BEFORE THE MACHINE WILL OPERATE. <06:03:58-29> (01:01:42-25) THIS PREVENTS
THE OPERATOR'S HANDS FROM BEING CAUGHT WHILE THE MACHINE IS CYCLING.
    <06:06:55-17> (01:01:46-03) THIS SEAM WELDER HAS AN ELECTRIC EYE.
<06:07:48-27> (01:01:49-10) WHEN A HAND OR ANY OTHER OBJECT IS INSERTED INTO
THE CAGE, THE BEAM IS BROKEN, STOPPING THE ROTATING TABLE. <06:06:56-18>
(01:01:55-29) THIS ALLOWS THE <06:07:23-27> (01:01:57-29) TABLE TO BE LOADED
WHILE PREVENTING ANY INJURIES FROM THE ROTATING ARM.
    <02:04:36-19> (01:02:00-28) MANY MACHINES HAVE A DOOR THAT MUST BE
CLOSED BEFORE THEY CAN BE OPERATED. <06:00:19-01> (01:02:05-04) THIS ONE HAS
A GUARD WHICH PREVENTS FLUIDS FROM BEING SPLASHED ON THE OPERATOR.
    <03:02:21-01> (01:02:08-28) ALL OVER THE PLANT YOU WILL FIND CAGES,
<06:17:31-01> (01:02:11-17) FENCES, <06:06:56-26> (01:02:12-11) SIGNS,
<06:12:21-24> (01:02:12-29) LINES, <02:07:24-19> (01:02:13-21) MIRRORS,
<06:03:55-17> (01:02:14-14) GATES, AND <06:10:34-29> (01:02:15-03) GUARDS,
ALL OF WHICH <10:04:44-05> (01:02:16-19) ARE DESIGNED TO KEEP YOU SAFE FROM
 INJURIES AND ACCIDENTS. PHOTOGRAPH SIGN COMING INTO PLANT - GATE 1 SAFETY
SIGN)
------------------------------IN-----------OUT--------DURATION----
 STOP   06:17:29-03       06:17:31-01   06:17:31-25   00:00:00-24
 STOP   01:02:00-06       01:02:11-17   01:02:12-11-  00:00:00-24
-----------------------------------------------------------------
```

**Figure** *3-A sample screen* dump *shows a typical script with time* stamps. *<06:17:31-01> indicates* that *a segment* from *tape number 06 starting at 17:31* frame 01 *is to be copied to (01:02:11-* 17) tape number *01 starting from 02: 11 frame 17 and stopping at (01:02:12-11) 02: 12 frame 11. The ending time code not only indicates the ending of the segment, it a/so is the beginning of the next segment.*

under the cursor, and it will delete the old time code and replace it with the new. This capability allows you to mark a tape location as many times as you like without entering a string of time codes.

### GETTING THE PICTURE

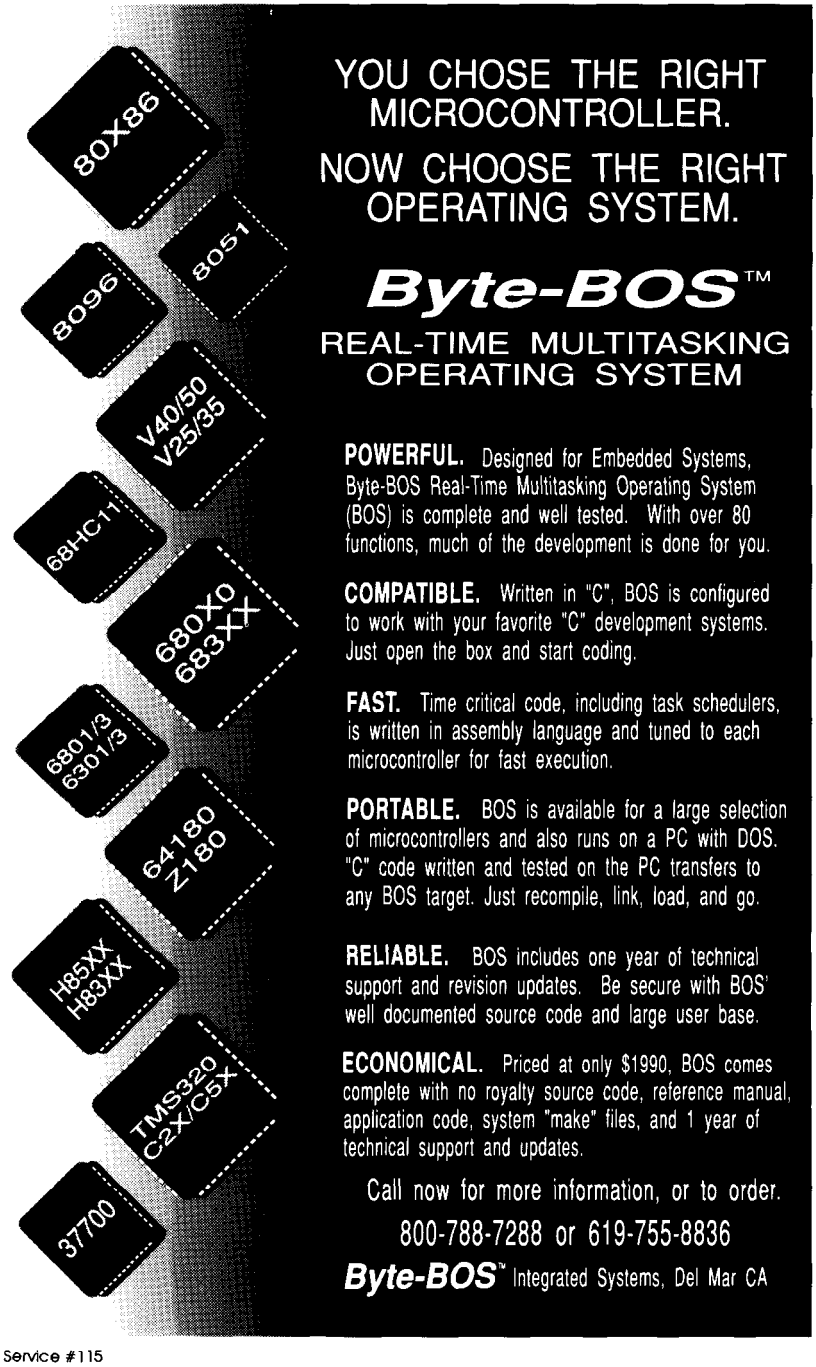
**Once you** have entered the source time code, you have all the elements you need to make an edit. The source and destination beginning time codes are still under the cursor, and the ending time code is the next edit location on the script. When you press <Ct rl L> the source and destination time codes are loaded into the command lines at the bottom of the screen. The next time code in the script is found and loaded as the cut-out point. The selected codes are highlighted in reverse video and are locked to the codes in the command line. Therefore, any changes made to the code on the command line will be reproduced on the script. You can now perform any of the edit commands listed. You can make trial edits and previews, and automatically record changes as you make them. When you're done, press <Ctrl U> to unlock the command line from the script, or press <Ctrl L> to unlock and lock onto the next pair.

After you have entered enough edit points, the software relates the current VCR tape time to the editing locations entered into the script as best it can. Pressing <P> scrolls the script and displays the computer's guess at the current tape time's location in the script. Pressing <Shift P> locks the screen display to the tape. You can play, rewind, or fast forward a tape and watch the script scroll in perfect sync. I haven't found any practical uses for this feature, but it's fun to watch.

### TAKE YOUR BEST SHOT

My method for logging work tapes arose while I was assembling "best-takes lists." What's a best-takes list? Well, when you go out on a "shoot" you don't come back with Shot #1 on Tape #1 and Shot #2 on Tape #2. Shots are never taken in order and are often retaken several times. It is far better to come back with more footage than you need, than to come up short, especially if you're in Michigan and your shoot's in Mississippi. I once did a video for the American Cancer Society that had 22 work tapes, each consisting of about 40 to 50 shots. I had about a thousand shots to keep track of. Large studios have people who do nothing but review work tapes. They pick the segments that are needed, copy them onto a best-takes reel, then pass them along to another person who does the editing. Free lancers usually don't have that luxury. When I assemble a best-takes reel, I log in a tape and then use that log to find the shots I need.

Once again the old ballpoint pen came into play. Logging a work tape consisted of viewing a segment, then jotting down a time code and a short notation of the scene. Because I had to take my eyes off the screen to write, I had to stop frequently and rewind. The whole process was cumbersome and time consuming. Well, the old Model IV came to the rescue again. The solution was to add rudimentary

Reader Service #115

```
EVENT #    9  IN:   01:02:05-04  SRC:   06:00:19-01
              OUT:   01:02:08-28
              DUR:   00:00:03-24
          PGM  LEN:  00:00:33-18

EVENT #   10  IN:   01:02:08-28  SRC:   03:02:21-01
              OUT:   01:02:11-17
              DUR:   00:00:02-19
          PGM  LEN:  00:00:36-07

EVENT #   11  IN:   01:02:11-17  SRC:   06:17:31-01
              OUT:   01:02:12-11
              DUR:   00:00:00-24
          PGM  LEN:  00:00:37-01

EVENT #   12  IN:   01:02:12-11  SRC:   06:06:56-26
              OUT:   01:02:12-29
              DUR:   00:00:00-18
          PGM  LEN:  00:00:37-19
-----------------------------IN------------OUT--------DURATION-----
 STOP   06:17:29-03        06:17:31-01   06:17:31-25   00:00:00-24
 STOP   01:02:00-06        01:02:11-17   01:02:12-11-  00:00:00-24
-------------------------------------------------------------------
```

Figure 4—*The Edit Decision List is a more conventional and easier to read method for displaying the control sequence.*

word-processing capability. Now, you might ask if I can type faster than I can write. I can't. But, I can enter a time code with one key press, and I don't have to take my eyes off the monitor. Best of all, my notes are legible when I'm finished. All in all, there is enough speed improvement that I rarely need to stop a tape. The software module that supports word processing is called WORD, which is invoked from the keyboard by pressing the </> key, and is designed for fast note taking and nothing else. When active, WORD is always in the insert mode. Going back and making changes can be done, but that's a real pain.

To use WORD, put the VCR into the play mode and then press </>. Originally, you used the three function keys of the Model IV to enter time code; now you're going to use them again, but this time with one important difference. As before, pressing <F3> enters the current tape time, but now the cursor automatically moves to the right of the time code instead of staying put, and it is ready for you to add notes. Enter notes as you would on any word processor. When you're finished with a note, press <Enter> to move to a new line that is ready for the next time code. Both the source and destination machines may be used in note taking. The <F2> key is for the source machine, and <F3> enters the time code for the destination machine. I generally use the <F3> key. <F1> pauses both machines in case you need more time.


# EXPRESS CIRCUITS

MANUFACTURERS OF PROTOTYPE PRINTED CIRCUITS FROM YOUR **CAD** DESIGNS

TURN AROUND TIMES AVAILABLE FROM 24 HRS — 2 WEEKS

**Special Support For:**

- TANGO.PCB
- TANGO SERIES II
- TANGO PLUS
- PROTEL AUTOTRAX
- PROTEL EASYTRAX
- smARTWORK
- HiWIRE-Plus
- HiWIRE II
- EE DESIGNER I
- EE DESIGNER III
- PADS - PCB
- ALL GERBER FORMATS

- FULL TIME MODEM
- GERBER PHOTO PLOTTING

NEW! WE CAN NOW WORK FROM YOUR EXISTING ARTWORK BY SCANNING. CALL FOR DETAILS!

Express Circuits

314 Cothren St., F!O. Box 58
Wilkesboro, NC 28697

Quotes:
I-800-426-5396
Phone: (919) 667-2100
Fax: (919) 667-0487

Reader Service # 142

## A PICTURE OF THE SOFTWARE

The software for the Model IV is written in machine code and is divided into eight modules.

MAIN handles the machine start-up functions. It contains the command interpreter for functions other than word processing. Upon start-up, MAIN issues a series of wake-up commands to the modules that ready them for use. It also handles some of the time code entry and delay timing for editing functions.

SUBS contains frequently used subroutines, including those used for transmitting edit information and machine commands to the modules, and the code for calculating edit durations.

DISKIO loads and saves files from and to disk. When you're queried for a file name you can type it in or press the right arrow key and let the computer leaf through the available files for you. DISKIO automatically adds the file extension ".VT" to the file name for easy identification.

EDITOR interprets basic machine commands. It's the real heart of the editing system and acts as an interface between you and the VCR modules. It interprets and translates your commands into something a VCR module can understand. EDITOR decides whether you are asking for a preview, an edit, or a review. It can also instruct the VCR modules to do a tape-to-tape edit or a camera-to-tape edit. It decides how long an edit will last and transmits that information to the VCRs. Finally, EDITOR contains the code necessary for videotape indexing.

INTERRUPT takes care of information coming from the VCRs. The actual interrupt routine is very short, only a few lines long (see Listing 1). INTERRUPT takes data from the serial stream emitted by the modules and inserts it into a ring buffer. After saving the data, it notifies the rest of the software that data has arrived in need of processing. When the computer has time, it polls to see if there is data to be processed. If there is, it picks the data out of the ring buffer and decides what to do.

INTERRUPT is optimized for speed. Instead of doing a linear series of tests to interpret the data, this module conducts branch testing. It checks for the most likely possibility: time code. If time code is in the process of being received, the module then stores it in a buffer. If no time code is found, INTERRUPT looks for commands in order of those most likely to occur. It also is responsible for updating the display with the latest information. While this module is reasonably fast, it can trip up if information is received too quickly. If you have trouble, slowing down the baud rate should help.

WORD handles the bulk of the word-processing functions. It's designed to handle basic input and nothing else. While it can be used for some word processing, it's really for quick note taking and other types of minor data entry. WORD follows the conventions set up for most other word processors on the Model IV. The arrow keys move the cursor right, left, up, and down; shifted arrows move it to top of file, bottom of file, end of line, or beginning of line. When a file is loaded from disk or typed in from the keyboard, WORD formats it by marking the end of each line of text with the removal of a space and the insertion of 80h. This marker eliminates the need for the program to reformat lines as the screen scrolls, freeing processor time for more demanding functions. The markers are removed when the file is saved to disk, so the text remains flat ASCII.

WORD also recognizes and manipulates embedded time codes, inserting them with the proper formatting. When deleting characters, WORD recognizes when the cursor is positioned over a valid time code and deletes the entire time code at once. WORD also handles time code trimming and maintains other screen information.

GLOBAL is a module that requires some explaining (see Listing 2). It contains no assembly language instructions as such, but it does contain instructions to the assembler for calculating screen locations. I needed this module because I often experimented with different display layouts during the development of the program. This experimentation was useful, but cumbersome and time consuming to implement. I needed to have all the screen references in one location to be easily tweaked, but the last thing I wanted was for the program to waste time calculating a reference point every time it had to write to the screen. My solution was to build instructions in the assembler to calculate screen locations and then load them where needed at assembly time.

The fastest way to do screen I/O on the Model IV is to build a work

```
; $INTER INTERRUPT - PUTS RECEIVED BIT INTO
;   RING BUFFER AND UPDATES POINTER

INTER   LD   HL, (RINGIN)       ; GET RING IN POINTER
        INC  L                  ; POINT TO NEXT
        LD   (HL),C             ; SAVE RECEIVED DATA
        LD   (RINGIN),HL        ; SAVE
        RET


; $INTERUP INTERRUPT ROUTINE
; DETERMINES MODE OF VCR
;   AND UPDATES TAPE NOW COUNTER

; FETCH DATA FROM RING COUNTER AND UPDATE DATA
; NOTE: NO REGISTERS SAVED USE CAUTION WHEN CALLING

INTERUP LD   DE, (RINGIN)       ; RING IN POINTER
        LD   HL, (RINGOUT)      ; RING OUT POINTER
        LD   A,L
        CP   E
        RET  Z                  ; RET IF NOTHING RECEIVED
```

**listing 1 -*To save processor* time, the interrupt routine is very short *and simply Inserts* receivedinformation into a ring buffer. Later the software polls to see if the "in pointer" *is different from the "out pointer." Ifthepointers are different, the incoming dataisprocessed and distributed.***

# New Low Cost In-Circuit Emulator

The DrylCE Plus is a modular emulator designed so you can get maximum flexibility from your emulator purchase. The base unit contains all the hardware necessary to support pods containing many of the most popular members of the 8051 family of embedded control microprocessors. Buy one base unit, and select one or all of the pods you need to do the job at a much reduced cost. You get the same great functionality found in our popular DrylCE 8031 emulator plus real-time Execute-to-Breakpoint, Line-by-Line Assembler, and much more. And the price is (almost) unbelievable! (Yes, it works with the Mac, too!)

Base Unit (w/RS-232 IF) -- **$299**
Available Now!

Available Pods:
8031/32, 80C31/32, 80C154, 80C451, 80C535, 80C552/562, 80C652, 80C51FA, 8751/52,87C51/52--**$149** each

16K Trace Buffer option: Avail. 1st Qtr '92
Standard 8031 DrylCE -- Still only **$199**
Enhanced 8031 DrylCE -- **$269**

## 8051 Simulation

The 8051 SIM software package speeds the development of 8051 family programs by allowing execution and debug without a target system. The 8051 SIMulator is a screen oriented, menu command driven program doubling as a great learning tool. **$99**.

## 8031/51 Single Board Computer

A fast and inexpensive way to implement an embedded controller. 8031/32 processor, 8+ parallel I/O, up to 2 RS232 serial ports, +5 volt operation. The development board option allows simple debugging of 8031151 family programs. **$99**ea

**Call for your custom product needs. Free Quote - Quick Response**

Other products available:
**MyGAL** - GAL Programmer $199
FORTH Card - FORTH development card for STD Bus $279 (OEM-$199)

VISA MasterCard AMERICAN EXPRESS Card

HTE
HiTech Equipment Corp
9400 Activity Road
San Diego, CA 92126
[FAX: (619) 530-1458]

(619) 566-1 892


```
        PSECT

        PUBLIC  SCREEN,COMLIN,TOP,MODLIN,DSPFOL,DSPSIN,DSPSOUT
        PUBLIC  DSPMIN,DSPMOUT,CURENT,BURENT,DSPDUR,DSPMST,DSPSST
        PUBLIC  DSPMMD,DSPSMD,NOLINES,DSPDURM,DCURENT,DBURENT
        PUBLIC  DDSPSST,DDSPMST,DDSPMMD,DDSPSMD

SCREEN  EQU     0F800H
WSCREENEQU      3000H   ; WORK SCREEN
NOLINESEQU      19      ; NUMBER OF LINES OF TEXT ON SCREEN

        ; SET COMLNE
LINE    DEFL    23
CHAR    DEFL    15
COMLIN  EQU     WSCREEN+(LINE*80)+CHAR

        ; SET MODLIN
LINE    DEFL    23
CHAR    DEFL    0
MODLIN  EQU     WSCREEN+(LINE*80)+CHAR

        ; SET DSPSIN
LINE    DEFL    21
CHAR    DEFL    28
DSPSIN  EQU     WSCREEN+(LINE*80)+CHAR

        ; SET DSPSOUT
LINE    DEFL    21
CHAR    DEFL    43
DSPSOUTEQU      WSCREEN+(LINE*80)+CHAR

        ; SET DSPMIN
LINE    DEFL    20
CHAR    DEFL    28
DSPMIN  EQU     WSCREEN+(LINE*80)+CHAR

        ; SET DSPMOUT
LINE    DEFL    20
CHAR    DEFL    43
DSPMOUTEQU      WSCREEN+(LINE*80)+CHAR

        ; SET DSPFOL
LINE    DEFL    22
CHAR    DEFL    3
DSPFOL  EQU     WSCREEN+(LINE*80)+CHAR

        ; SET TOP OF SCREEN
LINE    DEFL    0
CHAR    DEFL    0
TOP     EQU     WSCREEN+(LINE*80)+CHAR

        ; SET SLAVE TAPE POINTER CURRENT
LINE    DEFL    21
CHAR    DEFL    9
CURENT  EQU     WSCREEN+(LINE*80)+CHAR
DCURENTEQU      (SCREEN+(80*LINE)+CHAR).XOR.0400H

        ; SET MASTER TAPE POINTER CURRENT
LINE    DEFL    20
CHAR    DEFL    9
BURENT  EQU     WSCREEN+(LINE*80)+CHAR
DBURENTEQU      (SCREEN+(80*LINE)+CHAR).XOR.0400H

        ; SET MASTER DURATION LOCATION
LINE    DEFL    20
CHAR    DEFL    58
DSPDURMEQU      WSCREEN+(LINE*80)+CHAR

        ; SET SLAVE DURATION LOCATION
LINE    DEFL    21
CHAR    DEFL    58
DSPDUR  EQU     WSCREEN+(LINE*80)+CHAR

        ; DISPLAY MASTER STATUS (PAUSE ETC.)
LINE    DEFL    20
CHAR    DEFL    21
DSPMST  EQU     WSCREEN+(80*LINE)+CHAR
DDSPMSTEQU      (SCREEN+(80*LINE)+CHAR).XOR.0400H
```

*(continued)*

Listing **2**—*Global is used for* experimentation *with screen displays without consuming valuable processor time. The assembler translates the instructions into screen references used by the program. For example, COMLIN sets the* ***screen position*** *for the* **comment line;** *in this case, line 23, column 15. Changing these references can position the comment line anywhere on the screen without requiring any processor overhead.*

```
        ; DISPLAY SLAVE STATUS (PAUSE ETC.)
LINE    DEFL    21
CHAR    DEFL    21
DSPSST  EQU     WSCREEN+(80*LINE)+CHAR
DDSPSSTEQU      (SCREEN+(80*LINE)+CHAR).XOR.0400H

        ; DISPLAY MASTER MODE (PLAY ETC.)
LINE    DEFL    20
CHAR    DEFL    0
DSPMMD  EQU     WSCREEN+(80*LINE)+CHAR
DDSPMMDEQU      (SCREEN+ (80*LINE)+CHAR).XOR.0400H

        ; DISPLAY SLAVE MODE (PLAY ETC.)
LINE    DEFL    21
CHAR    DEFL    0
DSPSMD  EQU     WSCREEN+(80*LINE)+CHAR
DDSPSMD EQU     (SCREEN+(80*LINE)+CHAR).XOR.0400H

        END
```

Listing P-continued

screen in memory and then switch it to the display when needed. GLOBAL fixes a start location for the work screen, then it sets a row and a column for each element to be displayed based on that reference. I can now move any element to any screen location simply by changing a variable in the GLOBAL file and reassembling the code.

In the module NONAME you'll find a hodgepodge of misfits, pieces, and recent developments. All the experimental work goes on here. You'll find the code for creating EDLs, locking and unlocking script time codes, and screen follows and searches. You'll also find some experimental segments. For example, the autedt function performs any automatic editing. Beginning at the cursor location, autedt extracts editing information from the script and then automatically executes it. You can use this routine to automatically edit an entire tape.

## READY TO FADE SOUND AND PICTURE OUT...

There are a number of new and inexpensive video tape formats available that give impressive results. With a little bit of skill and knowledge you should be able to adapt the modules to control many different VCRs. Whether you're an amateur or a professional, I hope I've given you some new ideas and a different direction to follow when you build your own video editing controller.

"Fade sound and picture.. .. Kiss black, and go to commercial."

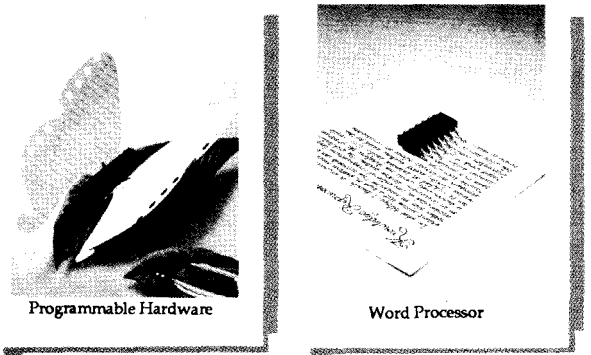
*Bill Kressbach holds a Master's degree in Instructional Media and Technology from the University of Toledo. When he's not shooting videos, he does some computer programming and is chief engineer* for a college *radio station.*

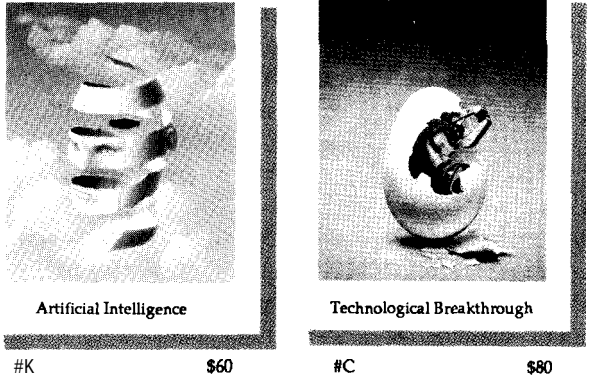
## IRS

**407** Very Useful
408 Moderately Useful
409 Not Useful


# *Limited Editions*

#G $60 #H $60




#K $60 #C $80

Circuit Cellar Ink cover artist Robert Tinney proudly offers these distinctive 16" x 20" Limited Edition Prints. Each is an exquisite reproduction from the pages of Byte Magazine, and is part of an edition of only 1000 prints: The museum quality stock is-acid free, ensuring brilliance and durability for decades to come. The artist personally inspects, signs and numbers each print, which is accompanied by its own Certificate of Authenticity.

Order your prints beautifully triple-matted and framed! The frames are of the silver metal variety, and mats are chosen to complement the colors of the print(s) you order. Plexiglass only.

The price of each print is shown at left. Order two or more and deduct 15%! Frames are only $39.50 each. For shipping, add $5 per order for unframed prints ($25 overseas); for framed prints, add $6.50 for one print and $4.50 for each extra print (ground). No frames shipped overseas. Full refund if not satisfied. For VISA, Mastercard or AMEX orders call 1-318-826-3003.

ORDER FORM cci

Qty. # Title Amount

If you order two or more, deduct 15% $
Frames ($39.50 each) $
Shipping charges: See above. $
Total $

☐ I have enclosed a check or money order to Robert Tinney Graphics. (Must be drawn on a US. bank; no foreign collection, please.)
Bill my ☐ VISA ☐ MasterCard ☐ American Express
Card #: Expires:
Name:
Address:
City: State: zip:
Country:

☐ Send a brochure showing your other prints.

ROBERT TINNEY GRAPHICS
P.O. Box 778
Washington, LA 70589

FEATURE ARTICLE

# High-Resolution Timing on A PC

Bruce Ackerman

In his article "A PC Stopwatch, Improved Timing for Acquisition and Control" (issue 19, page 22), David Schulze wrote that often PCs are incapable of high-resolution timing in a control or instrumentation project. Normally, timing is approached using either software loops or the tick timer, Software loops need to be calibrated against the tick timer before they are used, and not only tie up the processor, but can be grossly inaccurate as well. The tick timer's advantage is its ability to interrupt your software, but it's coarse by itself and in need of refining. I will explain these refinements, which extract a highly accurate timer with resolution greater than 1 μs from any PC, XT, or AT, with almost no additional system overhead and no additional hardware.

## THE TICK TIMER

The Intel 8253 (8254 for the AT) programmable interval timer is the source of the clock ticks. The 8253 has three independent channels that perform various rate divisions or other functions on its 1.19318MHz clock. Channel 0 generates the timer ticks, channel 1 clocks the dynamic RAM refresh, and channel 2 can be connected to the speaker.

Channel 0, utilized in this project, is normally set in the 8253's "mode 3," which is a frequency divider producing a symmetrical square wave. The 8253 has a 16-bit countdown register that causes the output pin to be high during the first half of the count and low during the second. The count register is initialized to 0, which acts like 65536, and is the factor by which the 1.19318-MHz clock is divided. The interrupt line is pulled by the 8253 output and generates 18.20648-Hz ticks.

The hardware interrupt vectors to the processor's interrupt 8, where the BIOS code resides to increment a 32-bit number in low memory (at location 0040:006C) and to handle disk-drive motor timing or any other chores a BIOS may have. The BIOS code also chains to a second interrupt, INT 1Ch, which usually contains only an `IRET` instruction. This 1Ch vector is the easiest to put your own interrupt handler on, should you want to be interrupted every tick. When your program exits (normally or abnormally), it must restore the original vector, or the system will crash.

## FASTER INTERRUPTS

At this point, you can increase the 18.2-Hz tick rate by simply changing the count register from 0 (65536) to some lower value. For example, a count of 59659 changes the tick timer to 20 Hz, which is handy simply because it's a whole number. (Of course, if we humans had 18.2 fingers.. .) This change will affect other software on the system, such as the time of day (your clock will run a little fast) and floppy disk drive timing (the time the motor is allowed to come to speed and the time it runs after an access), but in my experience the change from 18.2 to 20 Hz has never caused a problem. You can correct the time of day after you return the tick timer to its normal rate at the end of a program: keep track of the number of ticks that occurred at the faster rate and decrement the BIOS counter appropriately. The `set 8253()` function shown in Listing 1 reprograms the tick rate.

What if you want even faster interrupts? Running the BIOS routine at 16 times the normal tick rate would be a bad practice; if you are writing code that will be on a variety of machines, you don't know what other

```
/*Sets the 8253 channel 0 tick interval, putting the channel in
  the standard mode 3, the same mode that the BIOS initializes
  it to. The tick interval is normally 0, which is treated as
  65536, for 18.2 Hz, but may be set to, for example, 59659 for
  a 20-Hz clock*/

void set8253(unsigned cnt)
{
asm mov al,036h
asm out 43h,al          /*control reg: mode 3*/
asm mov ax,cnt
asm out 40h,al
asm mov al,ah
asm out 40h,al          /*16-bit counter*/
}
```

Listing 1 **-Reprogramming** *the 8253 for a new tick interval is very straightforward.*

things (e.g., battery management, TSRs) may be affected. At the very least, the time of day would start to look ridiculous. Therefore, in order to get the interrupt rate this high, program the 8253 for 16 times the normal rate (65536/16 = 4096), and install an interrupt handler on the INT 08 vector (not the 1C secondary vector). The handler performs the desired function at this high speed, and then chains to the BIOS routine every sixteenth call, with a far jump to the original location of the INT 08 vector. On the other 15 calls, simply reset the interrupt controller (`MOV AL,20h; OUT 20h,AL`) anddoan `IRET`.

```
#pragma inline

/*Sets the 8253 channel 0 to use mode 2, so the count value only
  counts down once per tick so we can read it unambiguously.*/

void timer_open(void)
{
asm mov al,034h
asm out 43h,al                  /*control reg: mode 2*/
asm xor al,al
asm out 40h,al                  /*count value 0 (65536)*/
asm out 40h,al
}

/*This function reads our 1.19318-MHz timer, assuming that we
  have called timer_init() beforehand to set it up.*/

unsigned long timer_read(void)
{
unsigned long count;

asm mov ax,0040h
asm mov es,ax
asm mov bx,es:006Ch         /*get BIOS counter*/

asm mov al,06h
asm out 43h,al              /*latch the count value*/
asm in al,40h               /*read LS byte*/
asm mov ah,al
asm in al,40h               /*read MS byte*/
asm xchg al,ah              /*put them where they belong*/
asm not ax
asm inc ax                  /*convert from upcount to downcount*/

asm mov cx,es:006Ch
asm cmp bx,cx
asm je rtn                /*if BIOS counter didn't change, done*/
asm cmp ax,8000h    /*decide which to use based on whether 8253*/
asm jb rtn              /*count was above or below its midpoint*/
asm mov cx,bx

rtn:
asm mov word ptr count,ax
asm mov word ptr count+2,cx
return count;
}

void timer_close(void)
{
asm mov al,036h
asm out 43h,al                  /*control reg: mode 3*/
asm xor al,al
asm out 40h,al                  /*count value 0 (65536)*/
asm out 40h,al
}
```

Listing *2-Using the 8253 in mode 2 allows more accurate intermediate count values to be* **read.**

## MUCH HIGHER RESOLUTION

There are limits to the clock rate that you can get by reprogramming the tick speed; an extremely high rate will load down the processor with interrupts. To get a truly high-resolution timer, you must take a different approach.

If you could read the current count from within the 8253 while it's counting, you would expect to see a 16-bit number that counts down at 1.19318 MHz. With the 8253 programmed in its default configuration so it counts through the full 65536, the BIOS tick counter in low memory serves as the 32 most-significant bits of a whopping 48-bit counter.

Alas, this technique has a problem (don't they all?). As I said earlier, the default "mode 3" operation of the 8253 channel 0 emits a symmetrical square wave: up for the first half of the countdown period and down for the second. The exact way the chip does this operation is curious. It counts down by two counts at a time rather than by one, holding the line high as it does; then it again counts down by two, this time holding the line low. While this way is fine for the 8253 to produce a symmetrical output, it causes a problem for us: when you look at the count, you'll see the chip count down by twos twice, but you'll have no way of knowing whether it's on the first or second half of the cycle.

You don't really need a symmetrical output, though! After all, the only function of the output line is to yank the interrupt periodically. So you can reprogram channel 0 in "mode 2," which only produces a low output for one period and then the other 65535 are high. This mode works fine for generating interrupts and uses a simple down-counter. The count starts at 0, wraps around to FFFFh,

```
#pragma inline

unsigned long hrt_count;            /*the 32-bit time*/

/*This function is the first one to call before using the 8253
  high-resolution timer. It sets the 8253 channel 0 touse mode
  2, and installs an interrupt handler on int 1C. Therefore the
  companion function hrt_close() *MUST* be called before the
  program exits!*/

void hrt_open(void)
{
asm mov al,034h
asm out 43h,al                      /*control reg: mode 2*/
asm xor al,al
asm out 40h,al                      /*count value 0 (65536)*/
asm out 40h,al

asm mov ax,351Ch
asm int 21h                         /*DOS get vector call*/
asm mov word ptr cs:oldvect,bx
asm mov word ptr cs:oldvect+2,es

/*Install one of our handlers, which one depending on whether
  the old handler was just an IRET or if it did anything more.
  If it was just IRET, we save time by not chaining back to it*/

asm mov dx,offset intruptl
asm cmp byte ptr es:[bx],0CFh       /*CFh is the IRET opcode*/
asm je use1
asm mov dx,offset intrupt2
use1:
asm push ds
asm mov ax,cs
asm mov ds,ax
asm mov ax,251Ch
asm int 21h                         /*set int 23 vector*/
asm pop ds
asm jmp short retn

/*Here are the interrupt handlers themselves*/

asm oldvect dd 0                /*storage for the old 1C vector*/
asm tickcnt dw 0                /*the 16-bit tick counter*/

asm intruptl:
asm inc cs:tickcnt
asm iret                            /*do our own return*/

asm intrupt2:
asm inc cs:tickcnt
asm jmp dword ptr cs:oldvect        /*chain to the old handler*/

retn: ;
}

/*This function which reads our 1.19318-MHz timer, assuming that
  we have called hrt_open() beforehand to set it up.*/

unsigned long hrt_read(void)
{
asm mov bx,cs:tickcnt               /*get initial tick reading*/

asm mov al,06h
asm out 43h,al                      /*latch the count value*/
asm in al,40h                       /*read LS byte*/
asm mov ah,al
asm in al,40h                       /*read MS byte*/
asm xchq al,ah                     /*put them where they belong*/
asm not ax
asm inc ax                /*convert from upcount to downcount*/

asm mov cx,cs:tickcnt
asm cmp bx,cx
asm je retn             /*if BIOS counter didn't change, done*/
asm cmp ax,8000h   /*decide which to use based on whether 8253*/
asm jb retn            /*count was above or below its midpoint*/
asm mov cx,bx

retn:
asm mov word ptr hrt_count,ax
asm mov word ptr hrt_count+2,cx
return hrt_count;
}
```

(continued)

**Listing** 3-A complete implementation of *the high-resolution timer using the interrupt handler technique.*

and proceeds back down to 0. As you read this count you can convert it to an up-counter from 0 to FFFFh.

Reading the 8253 on the fly is actually easier than it sounds because the chip has a command to latch the current count and return it, one byte at a time, on the next two reads. Note that the counter may turn over as you read, generating an interrupt. You must be careful to read the BIOS tick counter both before and after you read the 8253, and make sure to choose the proper value if you find them different.

The code shown in Listing 2 implements this scheme. I have written it for Borland's Turbo C, using the in-line assembler capability of that compiler. Calling timer_open() puts channel 0 in "mode 2." At any time thereafter, a call to timer_read() returns a 32-bit number. This number, derived from the low 16 bits of the BIOS timer count and from the 8253, appears as a 32-bit up-counter running at 1.19318 MHz. Before your program exits, call timer-close() to put things back as you found them, although if you don't it probably won't be noticed.

## THE MIDNIGHT BUG

The BIOS tick timer, as every avid reader of IBM documentation knows, resets itself every midnight. This assumes that the clock is set correctly; if not, "midnight" occurs whenever the BIOS count reaches the number of ticks there are in a day.

Many timing schemes on the PC are vulnerable to this midnight time warp. If you are using the high-resolution timer when this unfortunate event occurs, you will get a reading that is about 10 seconds off. I looked at work-arounds for this code and decided that most of them were too time-consuming at the high-resolution timer's reading point. You don't want to be doing BIOS calls or excessively complex code at this time because it will throw your readings off. There's little use having a 1-µs timer that takes 100 µs to sample.

The one fix that does not load the system appreciably is ignoring the

```
/*This function MUST be called before the program exits, after
  all use of the high-resolution timer is completed, to
  uninstall our interrupt handler from the 1C vector*/

void hrt_close(void)
{
asm mov al,036h
asm out 43h,al                    /*control reg: mode 3*/
asm xor al,al
asm out 40h,al                    /*count value 0(65536)*/
asm out 40h,al

asm push ds
asm lds dx,cs:oldvect
asm test dx,dx          /*if dx is 0, ours was never installed*/
asm jz skipit
asm mov ax,251Ch
asm int 21h                       /*DOS reset the vector*/
skipit: asm pop ds
}
```

**Listing S-continued**

BIOS tick count entirely and implementing your own tick counter! Install your own interrupt handler on INT 1Ch, the secondary tick interrupt, which increments a 16-bit number in your own code segment. This number is the one used for the high 16 bits when you read your high-resolution timer.

Being somewhat obsessed with efficiency, especially in this situation where I might be using the timer for a system profiler and I want to avoid any extra load on the system, I've even gone to the trouble of optimizing the interrupt handler a bit. When you install the handler, look at the address pointed to by the original 1Ch vector. If the first byte of that code is an IRET instruction, then you know that you're at the end of your 1Ch handler, and you don't need to jump back to the original handler. In this case, you just finish up with your own IRET, which does the same thing.

## A COMPLETE EXAMPLE

Listing 3 is a complete implementation using the interrupt handler technique. Remember, your system is in danger of crashing if you don't include a provision to remove the interrupt handler on any sort of exit, normal or abnormal. The easiest way to clean up your alterations after the program finishes is by installing a dummy handler at the INT 23h vector, which is where the DOS code that aborts your program on Ctrl-C resides. The handler prevents the user from aborting your program, which gives you control over how the machine will be left when you're finished.

If you want to see an example of this Ctrl-C trapping technique, as well as play with the high-resolution timer, download the code for this article from the Circuit Cellar BBS. *[**Editor's Note:** Software for this article is available from the Circuit Cellar BBS and on Software On **Disk** #24. **See** page 105 for downloading and ordering information.]*

The 32-bit count rolls over approximately every hour. The timer also works fine for intervals less than one hour because the subtraction of two count values yields the correct interval even if it goes around a rollover. Treat the 32-bit numbers as unsigned when subtracting them.

Too bad we aren't counting a nice round rate like 1 MHz, but with this high resolution, converting the intervals you wish to measure from microseconds to the rate at which this timer runs isn't hard. For example, look for 1193 counts when you're timing a millisecond.

*Bruce Ackerman specializes in instrumentation and process control, writing software and designing hardware. He studied Computer Science at MIT and now works in research and manufacturing environments as well as product design.*

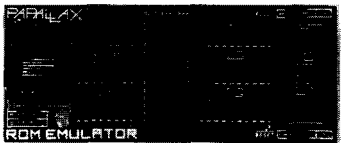
**IRS**

410 Very Useful
411 Moderately Useful
412 Not Useful


# EPROM EMULATORS

**An EPROM emulator appears as an EPROM to a target system. Instead of programming EPROMs, you simply download your code to the emulator. In seconds, you see results.**

## 27256 EPROM EMULATOR




**Emulates 2764, 27128, & 27258 EPROMs.**

Plugs into target EPROM socket and connects to PC parallel port via telephone cable.

Loads Intel, Motorola, hex, and binary files.

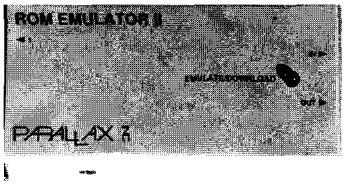
Reset outputs restart target after downloading.

Downloads 32K in 2 sec. (12 MHz PC AT)

**$199**

## 27010 EPROM EMULATOR

**Up to 4 units can be daisy-chained to emulate consecutive EPROMs and to support 16 and 32-bit systems.**




**Emulates 2784, 27128, 27258, 27512, and 27010 EPROMs.**

Plugs into target EPROM socket and connects to PC parallel port via telephone cable.

Reset outputs restart target sytem.

**$349**

VISA PARALLAX MasterCard

**(916) 721-8217**
**Fax (916) 726-1905**

Parallax, Inc. • 6200 Desimone Lane, #69A
Citrus Heights, CA 95621

**Dealer/OEM Pricing Available**

Prices subject to change without notice.
California residents add appropriate sales tax.
Add $4.00 for UPS ground, $7.00 for UPS 2nd day,
$15.00 for UPS next day.

Reader Service # 178

SPECIAL SECTION

# The K-100 MC68HC11-based Single-Board Computer

*Frank Swiger*
*& Joe Glover*

*On-board keyboard and video interfaces complete this single-board powerhouse*

**You've *heard* the *story before; you have a design that requires a computer and some I/O. Do you use off-the-shelf components to minimize design time and expense, possibly compromising on features, of do you design something specifically suited to the tusk?***

We recently had a job to put together a custom communication and control system. Typically, such a system utilizes a PC and five or six boards. Even with dropping PC prices, we still felt there had to be a cost-effective alternative to all this hardware. So we decided to look for an inexpensive microcomputer that would function more efficiently. When we started looking through ads, we soon realized finding the right processor was going to be tough. Most boards required a terminal or didn't have the features we needed. Sometimes you just have to build your own.

The search for a suitable microprocessor on which to base our board brought us to Motorola and their MC68HC11EVB evaluation board. What impressed us about the MC68HC11 were its features and the extensive amount of development software available free from Motorola. We chose MC68HC11F1 chip over the MC68HC11A1 because it is nonmultiplexed, has more internal RAM (1K), has eight additional I/O pins, plus has all the other standard goodies.

Our single-board solution, dubbed the FS-100, includes the MC68HC11F1 processor, one RS-232 serial port, 64K of RAM, up to 32K of EPROM, an interface for a standard IBM PC-type keyboard, a TTL monochrome video interface, and a STD bus interface should expansion be necessary in the future. Figure 1 shows a block diagram of the board. Figure 2 shows a complete schematic for the board.

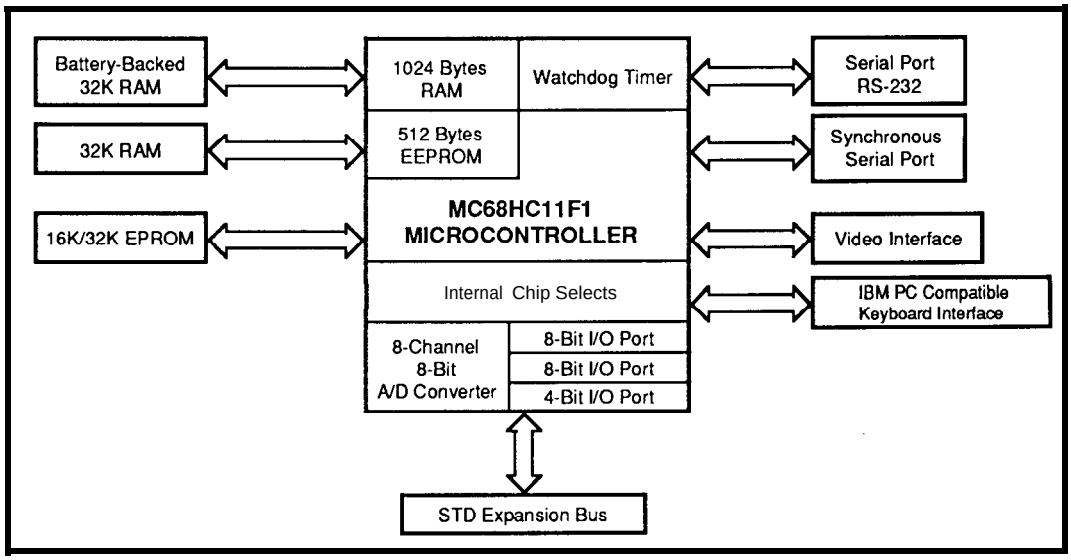



**Figure 1** — *The FS-100 single-board computer includes a Motorola MC68HC11F1 processor, RS-232 serial port, 64K of RAM, up to 32K of EPROM, IBM PC-type keyboard interface. TTL monochrome video interface, and STD bus interface.*

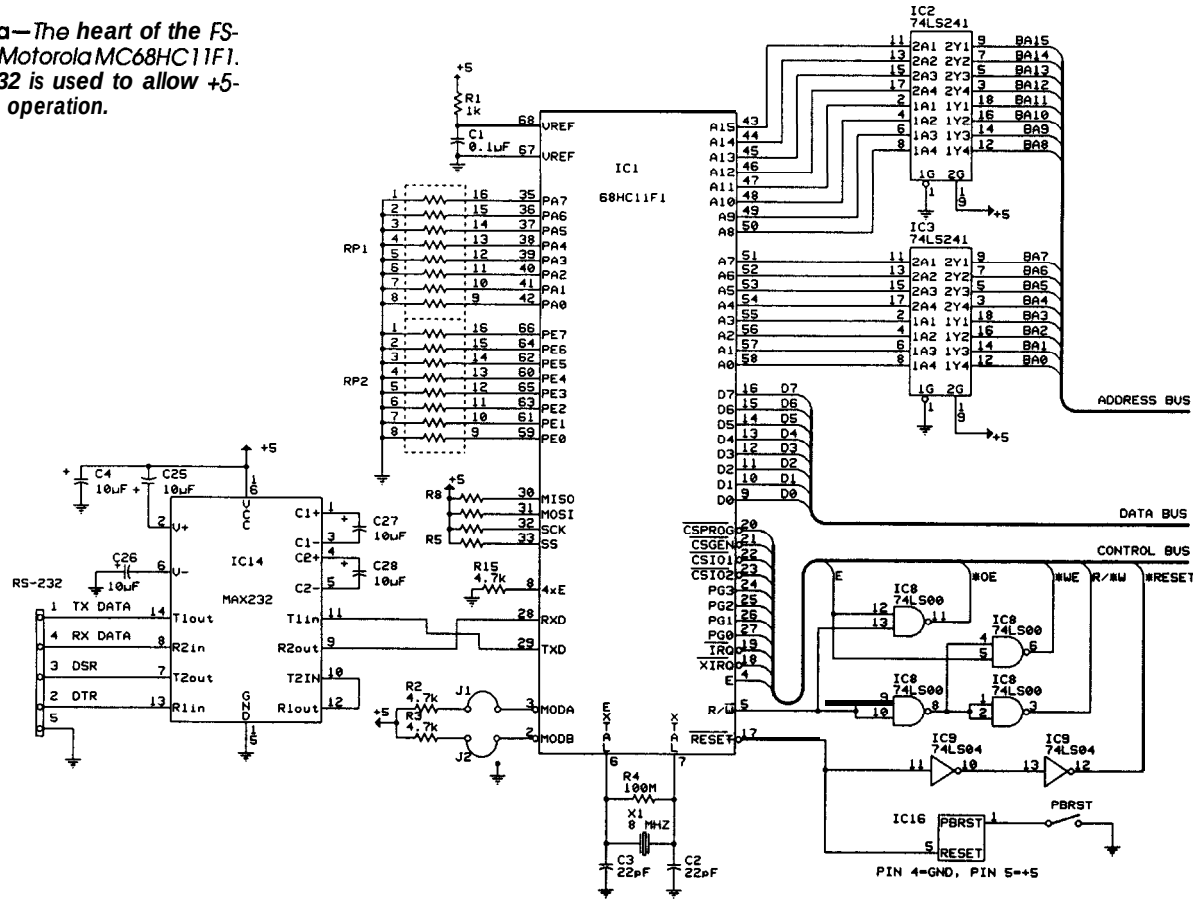



**Figure 2a**—*The heart of the FS-100 is the Motorola MC68HC11F1. A MAX232 is used to allow +5-volt-only operation.*

## MC68HC11 PROCESSOR

The Motorola MC68HC11 microprocessor has proven itself time and time again in the embedded application arena as a highly capable player. Figure 3 summarizes the chip's features. In their article "Using the Motorola MC68HC11" in issue #18 of CIRCUIT CELLAR INK, Steve Ciarcia and Burt Brown did an excellent job of describing the processor, so we won't rehash it here.

Some elaboration of how the chip's ports are used is needed. In its expanded, nonmultiplexed mode, our MCU can address up to 64K bytes of address space. Port F is for high-order address bits output. Port C is where the bidirectional data bus appears. Port E can be used for general-purpose static inputs, A/D channel inputs, or both. Ports A, D, and G can be used for general-purpose I/O, although each has an alternate function. For example, port A bits control timer functions and port D handles

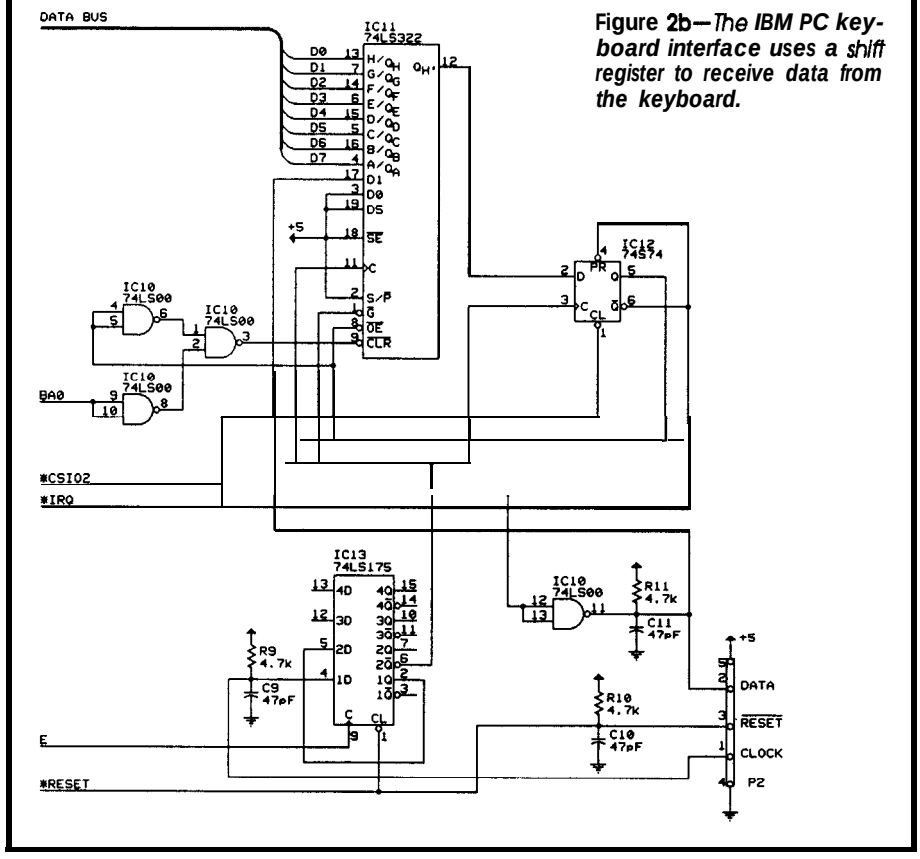



**Figure 2b**—*The IBM PC keyboard interface uses a shift register to receive data from the keyboard.*

serial I/O. Also, ports A and E are terminated with resistor packs that can be replaced with 0.1" header pins for external use.

Total board power consumption is 300 mA. The IS-100 runs a 2-MHz MC68HC11F1, however Motorola now has a 4-MHz version available.

## MEMORY

The FS-100's main RAM is mapped at $0000–$7FFF and its ROM at $8000–$FFFF. An additional 32K of optionally battery-backed RAM is mapped into the $0000–$7FFF range and used as a RAM disk. BASIC11's ESAVE and ELOAD commands, which are used to save and reload autostart programs, were modified to use the RAM disk. Either a 32K x 8-bit RAM or a Dallas DS1230AB battery-backed RAM can be used.

The MC68HC11F1 has four programmable chip selects: two for I/O (CS101 for keyboard and CS102 for video), one for main RAM (CSGEN), and one for ROM (CSPROG). The chip select control register enables the chips so they don't conflict with each other or with the processor's internal memory. The processor's PG2 output is used to select between the main RAM and the RAM disk.

## KEYBOARD AND VIDEO

With cost effectiveness still in mind, we provide interfaces for a standard IBM PC-type keyboard and a TTL-type monochrome monitor. The keyboard generates scan codes that must be converted by software to ASCII codes, as shown in Listing 1. We've made no attempt to fully support all the function keys and other extras supported by the keyboard. Feel free to expand on the software if such features are important to you.

The keyboard sends the scan codes back to the computer synchronously using a data line and a clock line. The data line feeds a shift register (IC11) clocked by the keyboard via IC13. When the last bit has been received, IC12 generates an interrupt (*IRQ), and the software puts the ASCII character in a RAM scratch pad. The interrupt and shift register are cleared when the data is read from the shift register.

We used the CRT9153 Enhanced Video Terminal Logic Controller from Standard Microsystems Corp. for video output. It is a programmable 40-pin chip that combines video timing, video attributes, alphanumeric generation, and block graphics generation (a block diagram is shown in Figure 4). This chip (IC6), coupled with a 2K x 8-bit static RAM (IC7), gave us the features we needed for our control system. Most importantly, we needed reverse video, intensity control, underline, and block-type graphics with four to six independently addressable segments per

Figure 2c—*The CRT9153* Enhanced *Terminallogic Controller makes adding a video* display to any processor a snap.

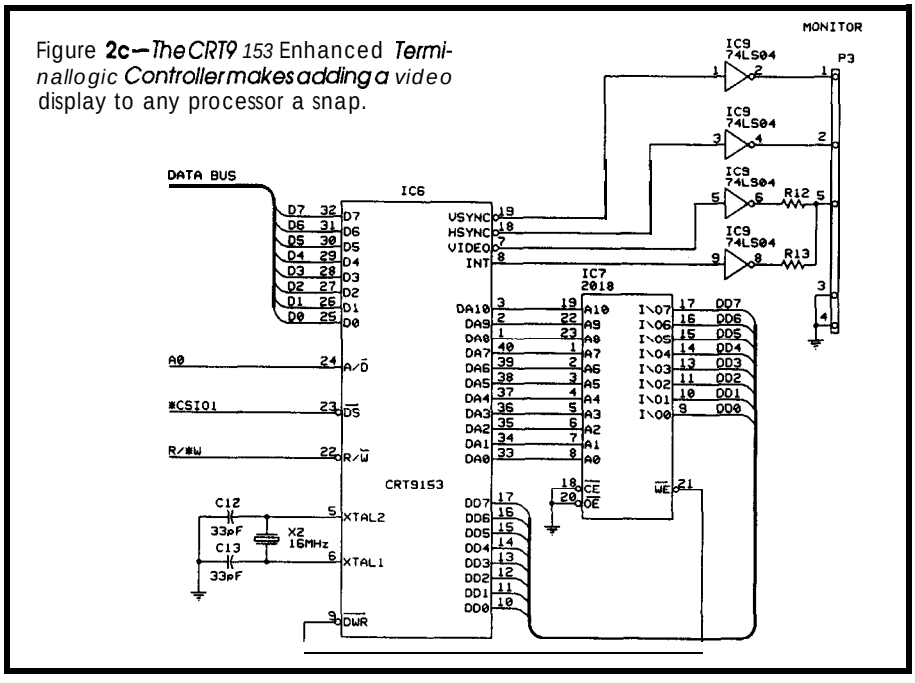
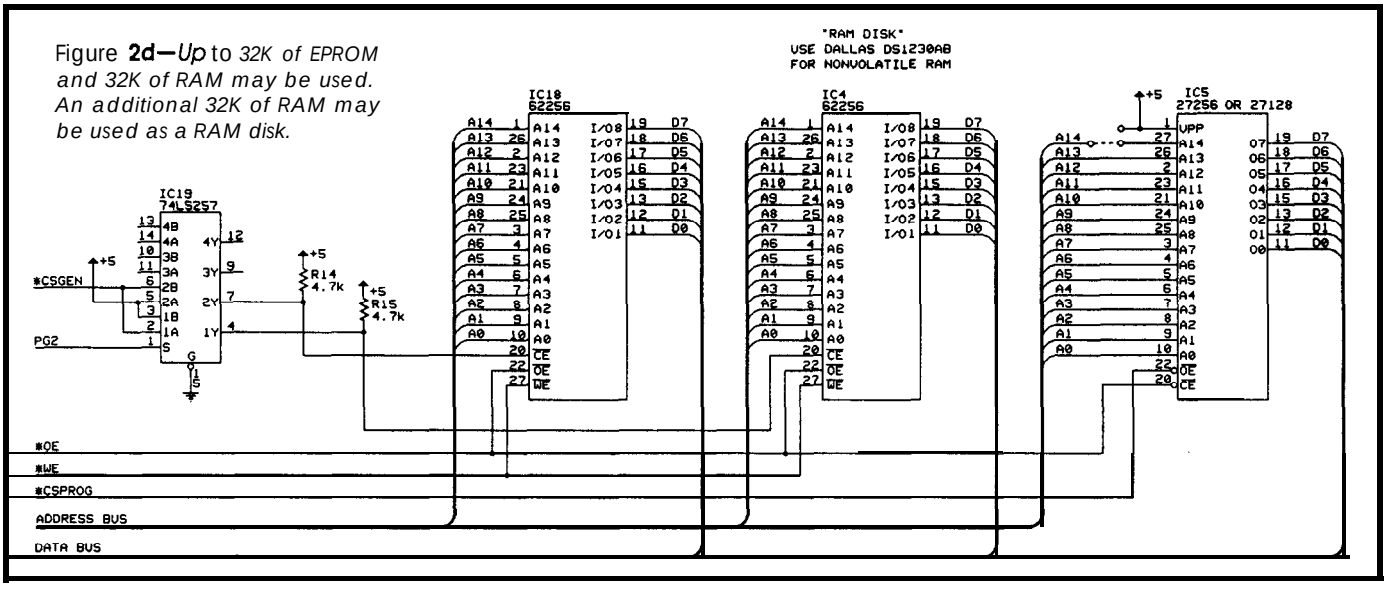



Figure 2d—*Up to 32K of EPROM and 32K of RAM may be used. An additional 32K of RAM may be used as a RAM disk.*



```
CLRKEY  EQU   $1800     a read here resets the interface
KEY     EQU   $1801     read keystroke data here

JIRQ    RMB   3         irq jump vector is put into RAM
KPAD    RMB   1         keyboard interface scratch pad

START   LDAA  #$7E      opcode for jump
        STAA  JIRQ      next 3 lines load jump vector
        LDX   #SCAN     for irq
        STX   JIRQ+1

        LDAA  CLRKEY    reset interface
        CLI             clear all interrupt flags

* The MPU is put in a loop,  waiting for irq from the interface

SCAN    LDAA  KEY       read key (on irq, program jumps here)
        STAA  KPAD
        ANDA  #$80      test for make or break code
        CMPA  #$80      is it break code?
        BNE   SCAN1     no it is make code branch to scan1
        LDAA  CLRKEY    yes it is a break code throw it away

        RTI             return from interrupt; go back to loop

SCAN1   LDAA  KPAD      get scan code
        LDX   #ATABLE
        ABX
        LDAA  0,X       convert to ascii
        STAA  KPAD      store ascii in scratch for later use

        RTI             return from interrupt; go back to loop

ATABLE  EQU   *
        FCC   " "
        FCC   " "
        FCC   "1"
        FCC   "2"
        FCC   "3"
        FCC   "4"
        FCC   "5"
        FCC   "6"
        FCC   "7"
        FCC   "8"
        FCC   "9"
        FCC   "0"
        FCC   "-"
        FCB   $7F
        FCB   $17
        FCC   " "
        FCC   "Q"
        FCC   "W"
        FCC   "E"
        FCC   "R"
        FCC   "T"
        FCC   "Y"
        FCC   "U"
        FCC   "I"
        FCC   "O"
        FCC   "P"
        FCC   "["
        FCC   "]"
        FCB   $0D
        FCB   $5E
    F   C   C "A"
        FCC   "S"
        FCC   "D"
        FCC   "F"
        FCC   "G"
        FCC   "H"
        FCC   "J"
        FCC   "K"
        FCC   "L"
        FCC   ";"
        FCC   "'"
        FCC   " "
        FCC   " "
        FCC   " "
        FCC   "Z"
        FCC   "X"
        FCC   "C"
        FCC   "V"
        FCC   "B"
        FCC   "N"
        FCC   "M"
        FCB   $18
        FCB   $06
        FCC   " "
        FCC   " "
```

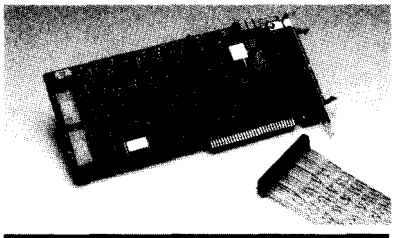
For lower case letters you would have to set up another table and write a routine that would shift between the two. BASIC11 automatically converts lower case to upper case, so foi our application lower case was not needed.

listing 1 — *The* scan ***codes from the IBM PC keyboard must be converted*** *to ASCII.*


**Real Time Devices**

*"Accessing the Analog World"*

***Quality U.S.-manufactured PC Bus cards and software for single user, OEM, or etnbedded applications.***



**AD3700 - $395**

*200* kHz THROUGHPUT

- 8 S.E. analog inputs, 12-bit 5 µsec A/D
- FIFO interface & DMA transfer
- *Trigger-in and trigger-out; pacer clock
- 4 Conversion modes & channel scan
- 4 Independent timer/counters
- 16 TTL/CMOS digital I/O lines
- *Assembler, BASIC, Pascal & C source code

**DataModule** PRODUCTS

*Plug-compatible with AmproCoreModule*

DM402 12-bit100 kHz analog I/O board with trigger, T/C, DMA &16 DIO lines $395

DM602 12-bit 4-channel D/A; voltage range select; current loop & DIO control $289

**DM802** 24-Line opto 22 compatible 82C55 PPI-based DIO interface ..... $149

POPULAR XT/AT PRODUCTS

AD1000 8 S.E. 12-bit A/D inputs; 25 kHz throughput; three 8-MHz timer/counters; 24 PPI-based digital I/O lines. ..... $275

ADA1100 AD1000 with 38 kHz throughput, 2 D/A outputs, and configurable gain .... $365

ADA2000 8 Diff./16 S.E. analog inputs; 12-bit 20 µs A/D; 12 or 8 µs A/D optional; two 12-bit D/A outputs; programmable gain; 3 T/Cs; 40 DIO lines from 82C55 PPI $489

ADA3100 8 Diff./S.E.12-bit analog inputs; 200 kHz throughput; gain select; FIFO interface & DMA transfer; pacer clock; external trigger; 4 conversion modes, multi-channel scan & channel burst; 4T/Cs; 16 DIO lines; two fast-settling analog outputs ... $659

AD.510 8 S.E. inputs; 12-bit integrating A/D with programmable gain ..... $259

ADA900 4 Diff./S.E. inputs; 18-bit V/F type A/D; variable resolution & conversion speed; 16-bit@ 16 Hz; 12-bit D/A, T/C &16 DIO lines $410

**DA600/DA700** Fast-settling 2/4/6/8 -channel 12-bit D/A; double buffered ..... **$192/359**

**DG24/48/72/96** Digital I/O lines; 82C55 based; optional buffers & line resistors **$110/256**

TC24 Am9513A System Timing &82C55 Digital I/O control card ..... $218

MX32 External analog multiplexer ..... **$198**

**ATLANTIS/PEGASUS/PEGASUS Acquire** Menu-driven, real-time monitoring, control, data acquisition and analysis turn-key software packages ..... **$150/290**

***Call for your Free Catalog!***

RTD logo, "Accessing the Analog World", and DataModule are trademarks of Real Time Devices, Inc. AMPRO and CoreModule are registered trademarks of Ampro Computers, Inc. opto 22 is a registered trademark of Opto 22, Inc.

**Custom/OEM designs on request!**

**Real Time Devices, Inc.**

*State College, PA USA*

rtd *Tel.: 814/234-8087*
*FAX: 814/234-5218*

Reader Service # 186

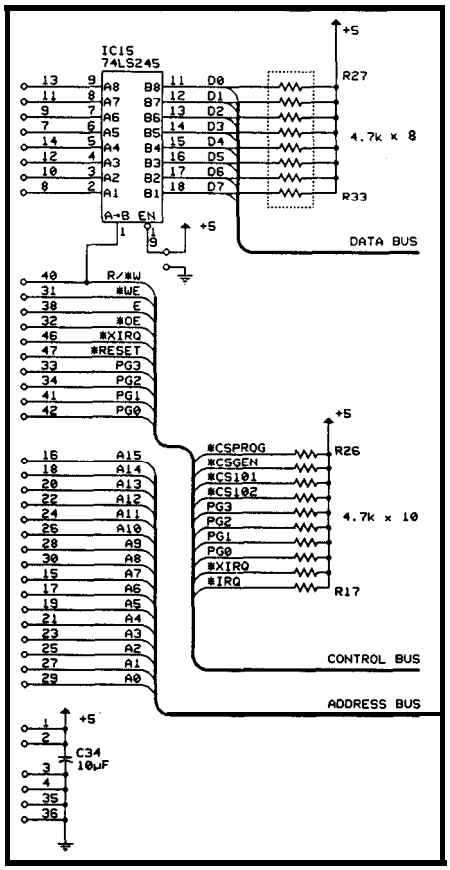
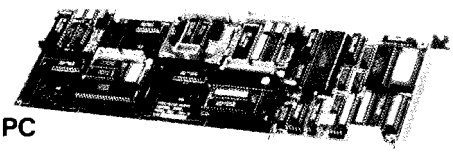



Figure 2e— The *FS-100 STD bus connections.*


*Position and/or Velocity*

# Motion Control



DCX-PC

## *8-Axis Programmable Motion Controller*

- State-of-the-art Digital **Multifunction** Controller can be configured in minutes using " off -the-shelf" components
- DCX motherboard contains intelligence/memory/firmware and 8 sockets for "plug-in" Modules
- 1 to 8 "plug-in" DC Servo, Stepper or Special Purpose Control Modules can be mixed/matched on same DCX board
- Install in any PC/XT/AT compatible, or use "stand-alone" with RS232 and/or IEEE-488 interface Modules
- High level interface libraries in " C" and " BASIC," with examples and source code, included

Precision Micro Control
CORPORATION
8122 Enoineer Road, San Diego, CA 92111
(619) 565-1500 FAX (619) 565-1511

Reader Service #182

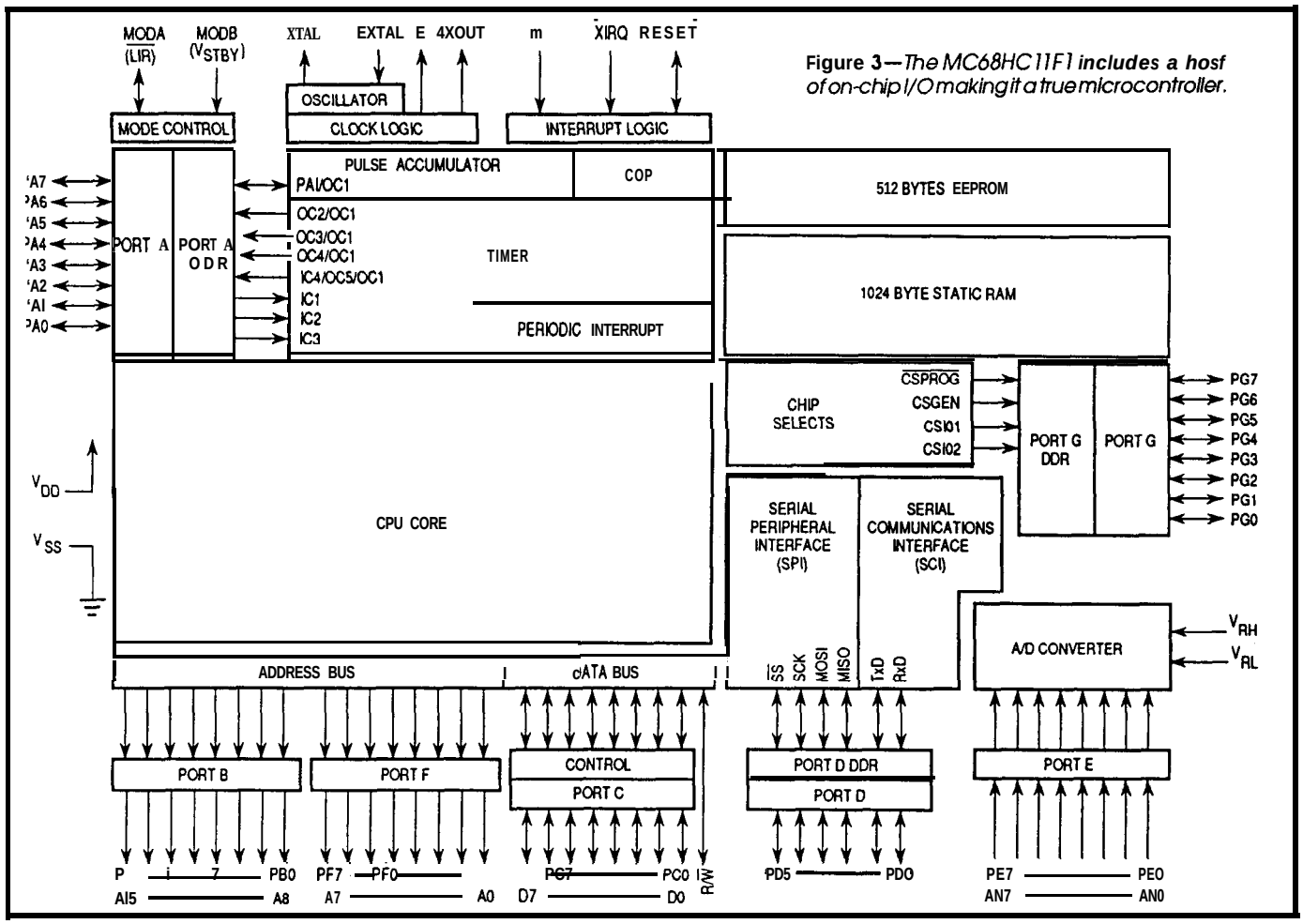




**Figure 3**—*The MC68HC11F1 includes a host of on-chip I/O making it a true microcontroller.*

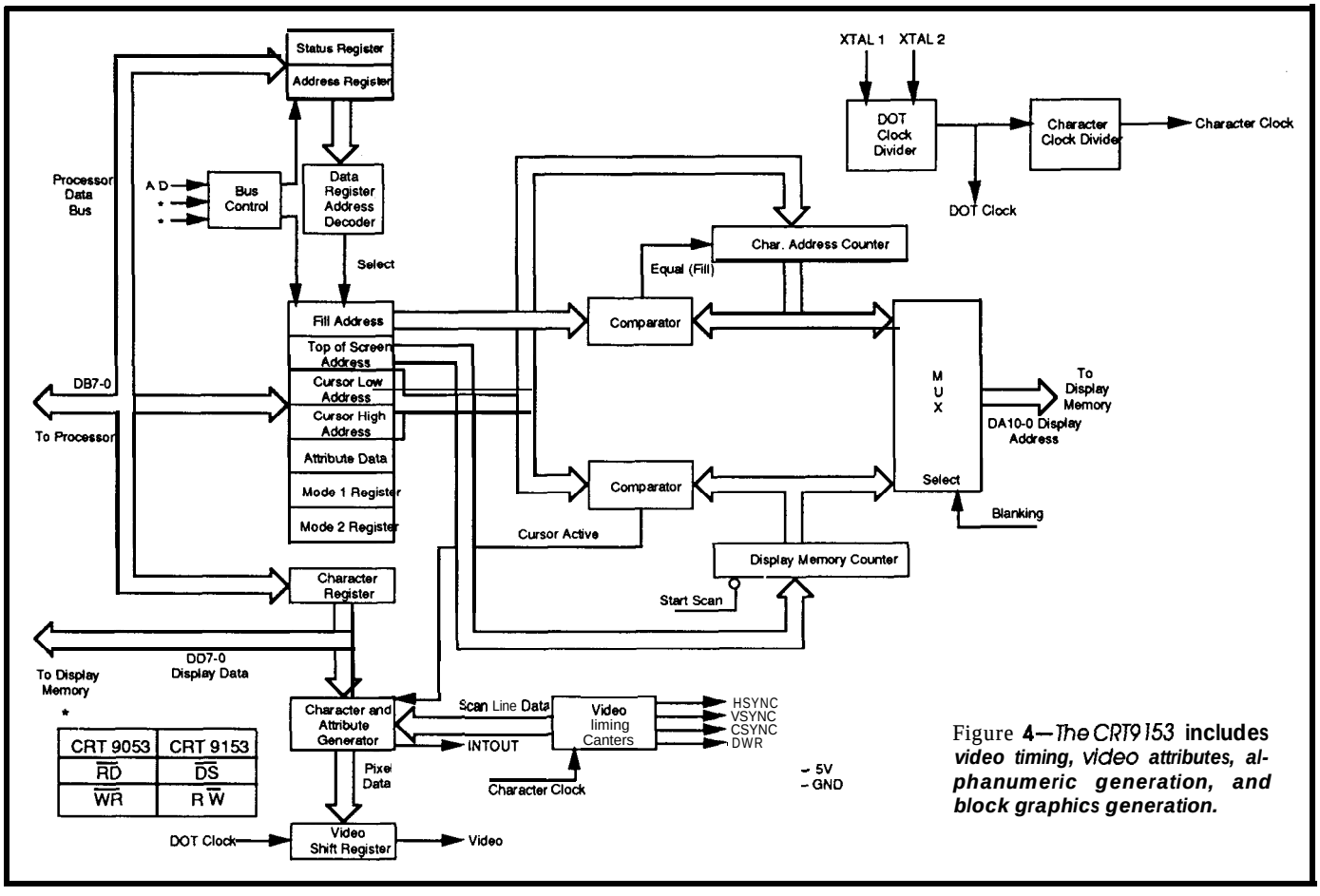
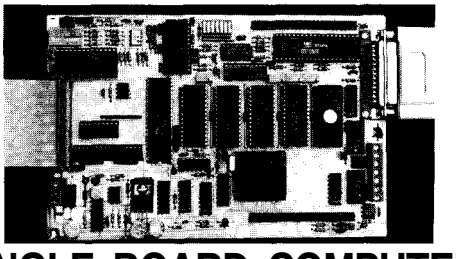



Figure 4—*The CRT9153* **includes *video timing, video attributes, alphanumeric generation, and block graphics generation.***


# INDUSTRIAL STRENGTH



## SINGLE BOARD COMPUTER

HAS OPTIMUM FEATURES FOR DATA AQUISITION AND CONTROL. PROFESSIONALLY DESIGNED AND BUILT FOR TOUGH APPLICATIONS WITH A 2 YEAR WARRANTEE AND A 30 DAY MONEY BACK GUARANTEE. PROGRAMMING IS A SNAP IN EITHER MULTI-TASKING BASIC, FORTH OR ANSI C. FEATURES INCLUDE: •

- 80PTICALLY COUPLED DIGITAL INPUTS
- 8 HIGH DRIVE BUFFERED DIGITAL OUPUTS
- 16 PROGRAMMABLE DIGITAL I/O LINES
- 16, THIRTEEN BIT ANALOG INPUTS
- 2 OPTIONAL TWELVE BIT ANALOG OUTPUTS
- 4 SERIAL RS232/485 SERIAL PORTS
- 6 TIMER/COUNTERS
- UP TO 512K MEMORY PLUS EEPROM
- E-PAC 3000 G2 COST $389.00 Q-W. 1

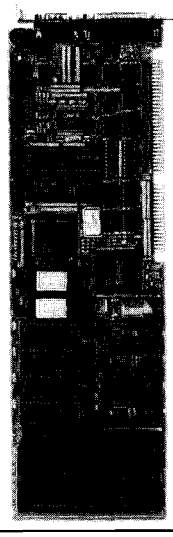
EMAC, inc.

618-529-4525 Fax 618-457-0110

P.O. BOX 2042, CARBONDALE, IL 62902

Reader Service #137





# IND-286 SBC

## AT Compatible DISKLESS SBC Includes DOS in ROM

**Complete** 16MHz 80C286 Single Board Computer for embedded PC applications features a 4M-byte PROMDISK disk emulator with battery back-up and an MS-DOS 3.3 compatible disk operating system in ROM.

*Features Include:*

- ❑ 4M-byte DRAM
- ❑ XT Size Board
- ❑ Keyboard Port
- ❑ 80287 Socket
- ❑ 2 COM, 1 LPT
- ❑ WatchDog Timer
- ❑ IDE Disk Port
- ❑ 4M PROMDISK
- ❑ Floppy Port
- ❑ 100% PC/AT Compatible
- ❑ Optional Video Daughter Bd.

Other Products:

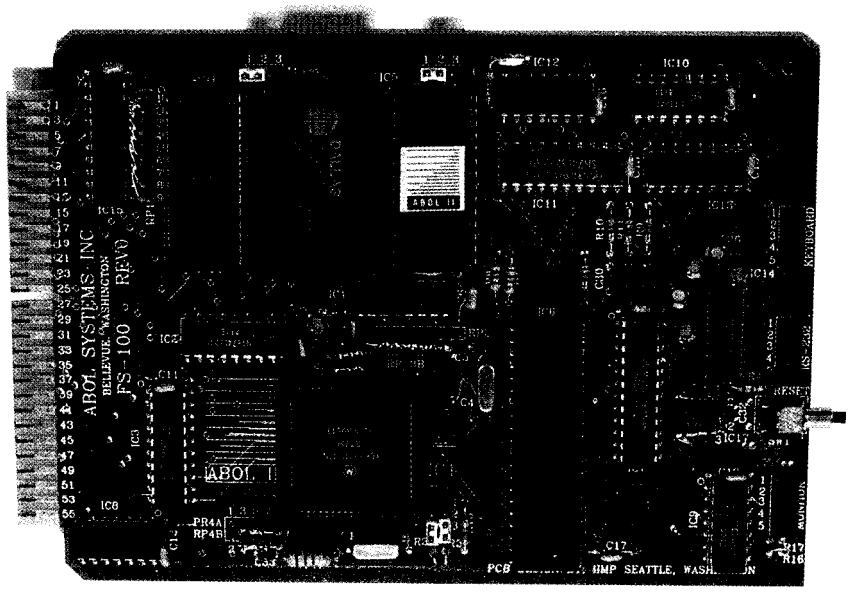
- IND-88 PC/ XT Single Board Computers
- PROMDISK III & IV Disk Emulators
- EPROM/RAM Memory Board
- FLASHDISK Driver for Micro Soft FFS
- FlexScan I & II Bar Code Decoders
- Custom PC Compatible Hardware & Software

micro computer specialists, inc.

2598-g fortune wdy vista, ca 92083

phone: 619/598-2177 fax: 619/598-2450

Reader Service #159




character space. We used a 17.1072-MHz crystal for the video, but any crystal between 16 and 18.7 MHz should work fine.

The CRT9153 also supports RS-170 composite video output, so if you're looking for a display controller for a future product, be sure to keep this chip in mind.

## SOFTWARE

Two pieces of software available from Motorola, the Buffalo monitor and BASIC1 1, are all that is necessary to develop working code for the FS-100. Both programs were described in detail by Steve and Burt in the article we mentioned earlier, so, again, we're not going to repeat the discussion here. Since the software from Motorola uses the MC68HC11's serial port, we had to modify the software to utilize the FS-100's on-board keyboard and video. We also modified BASIC11's ESAVE and ELOAD commands to use the RAM disk, as we mentioned earlier.

## CONCLUSION

The idea of a single-board computer is not new. However, it is hard to find one that will do all the tasks you require. After the best part of two years, our efforts resulted in the FS-100, which will serve as the heart of our communication projects and many other applications. Its small size and unique features make it perfect for use as a stand-alone data logger, home automation controller, and many other custom projects.

For more information about the FS-100, contact ABOL Systems Inc., 13240 Northup Way, Suite 21, Bellevue, WA 98005, (206) 747-8400.

*Frank Swiger is the manager of the ABOL Systems Inc. Portland, Oregon office. He is involved in the design of hardware and software for data communication systems and point-of-saleequipment.@ Glover is the Vice President of ABOL Systems.*

## IRS

413 Very Useful
414 Moderately Useful
415 Not Useful

**Cross-Assemblers** from $50.00
**Simulators** from $100.00
**Cross-Disassemblers** from $100.00
**Developer Packages**
from $200.00 (a $50.00 Savings)

**Make Programming Easy**
Our Macro Cross-assemblers are easy to use. With powerful conditional assembly and unlimited include files.

**Get It Debugged--FAST**
Don't wait until the hardware is finished. Debug your software with our Simulators.

**Recover Lost Source!**
Our line of disassemblers can help you re-create the original assembly language source.

**Thousands Of Satisfied Customers Worldwide**
PseudoCorp has been providing quality solutions for microprocessor problems since 1985.

**Processors**

Motorola 6800 RCA 1802,05 Intel 8051 Intel 8096,196kc
Motorola 6801 Motorola 68HC11 Motorola 6805
Hitachi 6301 Motorola 6809 MOS Tech 6502 WDC 65C02
Rockwell 65C02 Intel 8080,85 Zilog Z80 NSC 800
Hitachi HD64180 Mot. 68k,8,10

New
Zilog 28 Zilog Super 8

- All products require an IBM PC or compatible.

For Information Or To Order Call:

**PseudoCorp**
716 Thimble Shoals Blvd, Suite E
Newport News, VA 23606
(804) 873-1947 FAX:(804)873-2154

Reader Service #183

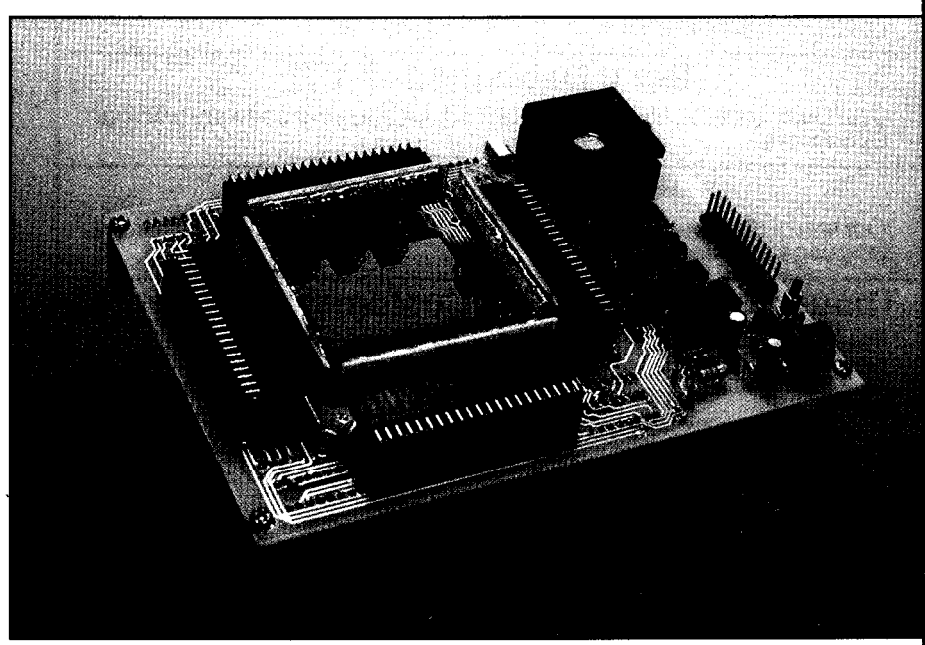
# 3rd DESIGN CONTEST RESULTS

One method we use at CIRCUIT CELLAR INK to measure our growth and success as a magazine is the ever-increasing response to our annual design contest. Our third year has not disappointed us. Once again, we are impressed with the quality and innovation of the designs submitted, which made our job of choosing winners extremely difficult. Please join us in congratulating all your fellow readers who made the Third Annual Circuit Cellar Design Contest a success, Several of our entrants and winners will be featured in upcoming issues or Volume II of The Circuit Cellar Project File. For those of you who believe you can rise to our challenge, find the on button for your slide-rule and start thinking about next year. But for now, let us salute the winners.

## OPEN CATEGORY

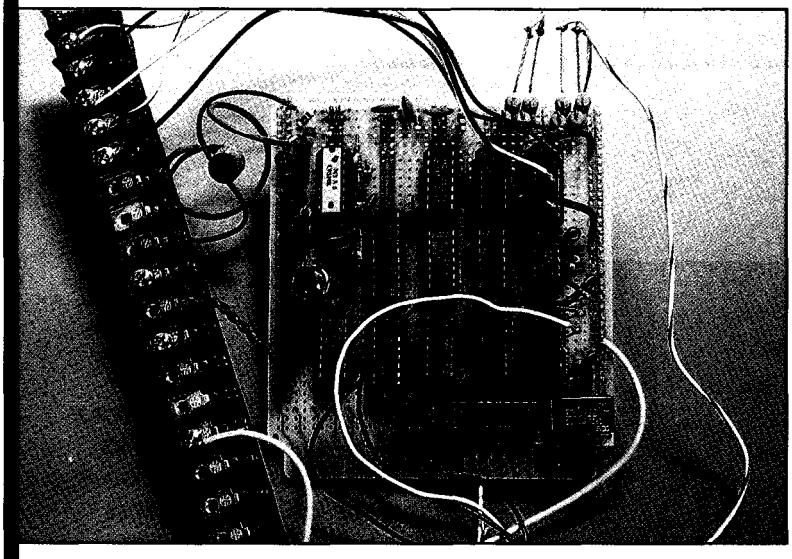
### First Place: $500

An MC68HC11-based Sensor for Two-Dimensional Position Measurement of Small Objects
by Brian Farmer



Brian's work in robotics led to his winning design of an MC68HC11-based sensor. He had programmed a robot to Insert geranium cuttings into peat plugs with preformed holes, Unfortunately, the robot functioned with strictly vertical motions and had no regard for the natural curvature to the plant cuttings. The result was a large amount of damage. So Brian designed a sensor that could be used in a robotic work-cell for measuring the extent of bend in plant material. The device uses LED and phototransistor arrays mounted on opposite sides within a square housing.
Brian's sensor is conditioned and serialized by dedicated circuitry, and it is controlled and monitored by an MC68HC11 microcontroller. The unique functionality of Brian's entry won our hearts and won him first place.

# COST-EFFECTIVE CATEGORY



## First Place: $500

The ScopeMotor, a Voice Operated Telescope Positioning System
by Michael Swartzendruber

Amateur astronomers are presented with a unique working environment: darkness. Adjustments to a telescope must be intuitively made by hand. Voice control would be the most logical answer, and Michael has constructed one such system that is extremely economical. The system has three major areas: a voice signal input and analog signal conditioning circuit. a voice recognition chip (the VCP-200) from Voice Control Products Inc., and a stepper motor clock source and high-current stepper motor drivers. The VCP-200 module allows true speaker-independent operation of most telescope positioning functions.

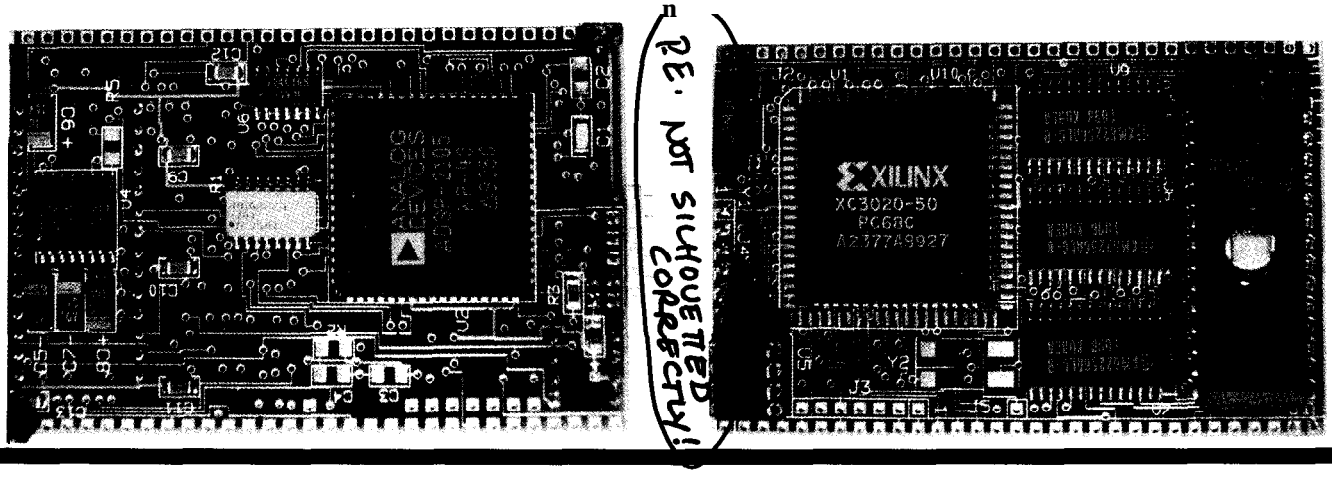
# DSP CATEGORY

Our new DSP Category apparently took people by surprise. As a result, we had only one entry that truly fit the category and was good enough to be a winner.

## First Place: Analog Devices EZ-LAB Demonstration Board, "Digital Signal Processing Applications," "Digital Signal Processing in VLSI," and $250

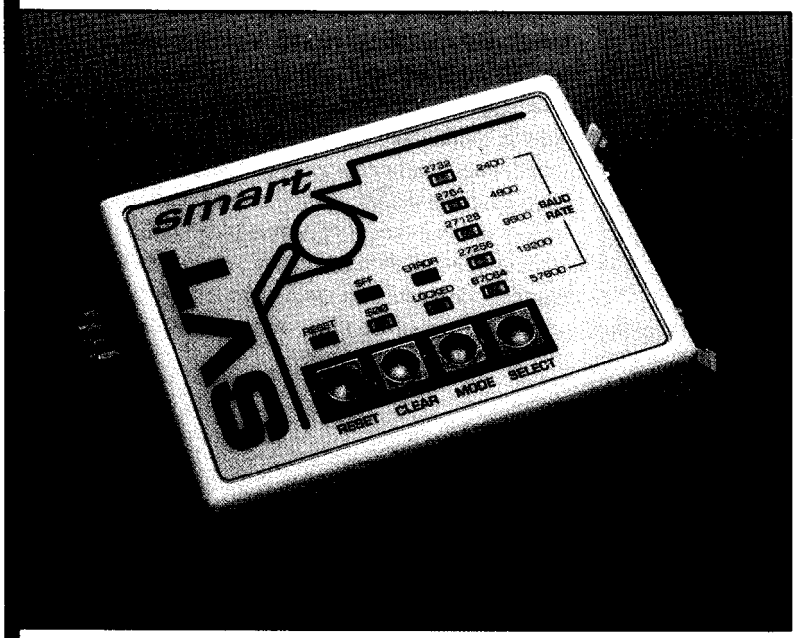
ADSP Module
by T. Bertinetti

Originating from France, this professional-quality, credit-card-size module uses the Analog Devices ADSP2105 along with a combination of on-board peripherals to make a very small but powerful digital signal processing system. Running at 10 MHz, the 6-layer surface-mount board includes up to 96K bytes of SRAM. up to 64K of EPROM, 2K bytes of EEPROM, a Xilinx programmable gate array, and an RS232 serial port. While designed for DSP applications, the ADSP Module is also well-suited to general embedded applications.



# 3rd DESIGN CONTEST RESULTS

## OPEN CATEGORY




### Second Place: $250

SmartROM
by Sanjaya Vatuk

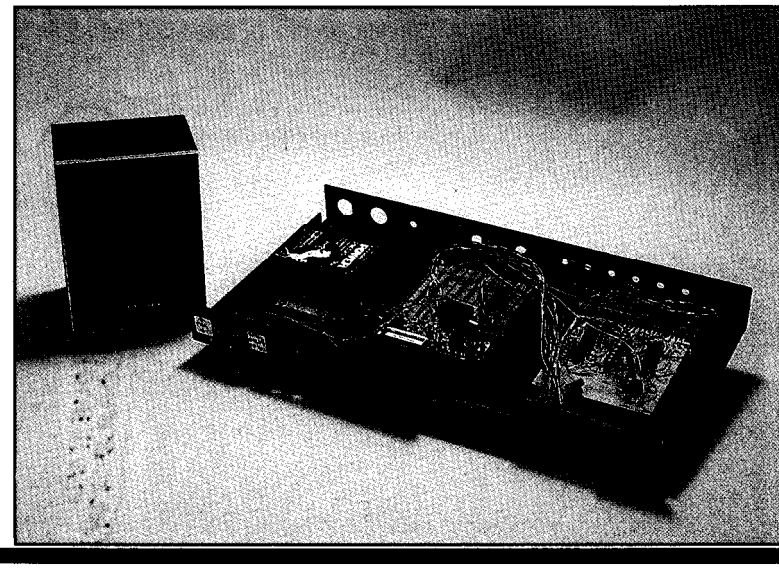
SmartROM, an intelligent EPROM emulator that needs only a standard RS-232 serial port as a host resource, is an exceptionally well-crafted device that uses either an 80C31 or an 80C32 microcontroller with firmware that can convert raw object files into binary EPROMs from 2732 to 27256. It can mimic the 87C64 EPROM with its on-board address latches. Sanjaya has built an emulator that is very fast, capable of downloading at most standard data rates up to 57,600 bps (115,200 bps using an 80C32) without any handshaking or delays. We were especially impressed with SmartROM's flexibility: a choice of six different data rates, file detection when downloading, status LEDs, and two options for blank state clinched second place for this project. Sanjaya also used ingenuity and creativity to give SmartROM professional-looking packaging using off-the-shelf materials.

## COST-EFFECTIVE CATEGORY

### Second Place: $250

An RMS Limiter
by Bob Gardner

Bob developed the RMS Limiter as an alternative to the expensive electronic speaker processors available that monitor signal levels and turn the volume down before the speakers can be damaged. A Motorola MC68HC11 is at the heart of this project, controlling a programmable attenuator circuit using input from a full-wave rectifier. Bob provides user feedback through a front-panel two-digit display.

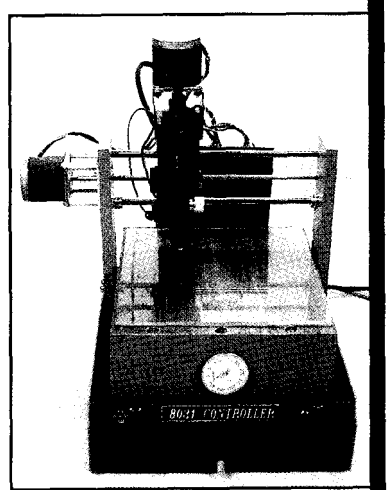


# OPEN CATEGORY

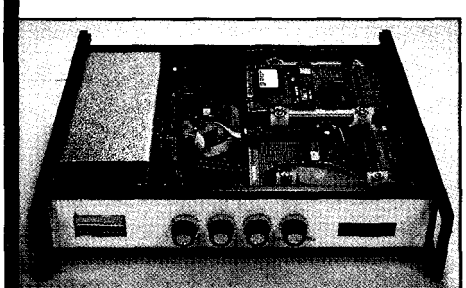
As we said earlier, the quality of our entrants was exceptional. The level of excellence led us to no other choice but to award three projects third place.

## Third Place: $100 (three awards listed in no particular order)

The PC-Controlled XYZ Table
by Alan Rauscher



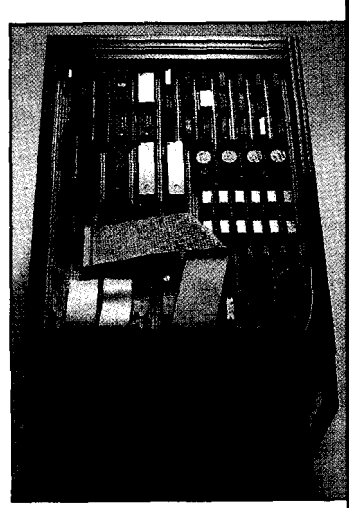

Alan used an IBM PC and an 8031 motor controller to achieve the extreme accuracy and repeatability that he needs when drilling printed circuit boards or panels He also wrote a TSR in order to use the table to engrave signs, panels, or name tags using a CAD package, such as AutoCAD or FastCAD. The Dremel drill motor can be moved In three directions: X, Y, and Z, with Z as the vertical movement of the drill turret. What intrigued us about the XYZ Table was the degree of accuracy obtained using simple materials readily available. Although the table was constructed of wood, the drill is accurate to five thousandths of an inch and is suitable for a board or panel of 6 x 6 inches.



The GPZ8. an Audio Digital Sampling System
by Tom Nickel

The GPZ8, an audio digital sampling system, is a audio effects generator created by Tom to be used with musical instruments, primarily the electric guitar. Tom built the GPZ8 around the Zilog Z80 microprocessor to simplify redesign of functionality, reprogramming of internal EPROM, and revising software. The enclosure is a rack-mountable 19" x 4" x 12" and is powered by 110 VAC. The GPZ8 digitizes a guitar signal, stores it, and then plays it back in either forward or reverse, adding delays or changing the speed to produce an assortment of echoes and sweeping reverberations. Specific effects can be chosen through two footswitches. Tom also added a 'fail-safe' circuit to protect a musician's performance should there be a malfunction or power difficulties.

PAWSDWC, A Power Arbitrary Waveform Synthesizer with Digital Waveform Capture
by Mark Wheeler

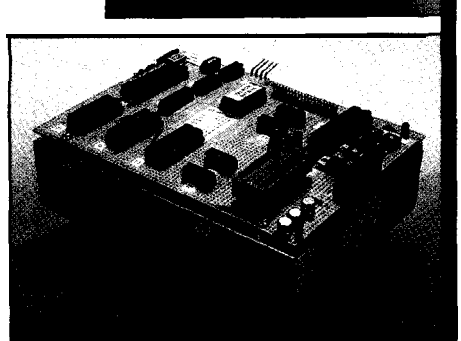


While the name is quite a mouthful (we defy anybody to pronounce it), we won't mumble when trying to describe this project. The list of supported functions would please anyone trying to work with analog signals: the PAWSDWC will capture and play back a waveform, plot the waveform on a printer using a standard Centronics parallel port, upload and download waveforms to and from an RS-232 port, allow hand-sketched waveforms to be digitized using a hand scanner for synthesis by the system, allow user control of input and output gains, and supply up to 40 volts peak-to-peak at 3 amps on its output. Mark intends to continue updating the system with new features in the future.

# COST-EFFECTIVE CATEGORY

## Third Place: $100

The Control Power of Embedded Forth
by Thomas R. Barnett



Forth is still the language of choice among die-hard embedded-application engineers, Thomas based his Forth Design Center on a version of Zilog's Z8 processor with a built-in forth interpreter. With the addition of some memory, a serial port, and an EPROM programmer, he put together a flexible system for doing forth program development.

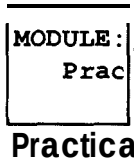
## DEPARTMENTS



page 66
Firmware Furnace


page 75
From the Bench


page 83
Silicon Update



page 91
Practical Algorithms


page 98
Domestic Automation


page 100
ConnecTime


# Micro-C, LEDs, and a Rotary Encoder

***Sturgeon's Rule has it that 90% of science fiction is junk. A similar rule applies to PC software, but I'd say that 90% is a lower bound Once in a while, though, you find a product that makes up for the rest.***

As longtime Furnace readers are well aware, I've been examining 8051 C programming for quite a while. Although I remain entirely underwhelmed by the tonnage of machine code per C instruction, I concede that C helps you produce better programs more quickly. As long as you keep the overall size of the project within reason, C is the way to go.

However, the cost of most 8051 C compilers and assemblers has been prohibitive for people who simply want to find out what all the fuss is about. That situation is now ended; the compiler I used for this project costs $50 and the companion assembler weighs in at $25. And best of all, they are excellent-quality tools that you need make no apologies for owning.

The project this time is an all-digital combination lock with an RS-232 output. The topic is certainly not original, but it's a convenient way to illustrate a C program that handles interrupts, updates an LED display, and works in 128 bytes of Internal RAM. In fact, the lock makes an ideal 8751 project because it uses only about 2K of code space (even in C!).

### AN HONEST LOCK

A standard combination lock, familiar to you from years of locking your bike to utility poles, has a digital dial. You twist it one way to set the first number, reverse course for the second number, and then back again to the final count. If you start out in the right direction and get all three numbers correct, the lock pops open. Of course, you can also slip a shim down the hasp and flip it open without knowing the combination, but that's another story.

The digital combination lock mimics that user interface, but improves on the display and overall security. Turning the knob updates one character in the display. Once you've got the right value, you pause and the next character lights up. Adjust all four correctly and the lock pops open.

The difference is the digital lock can't be shimmed. The four-character combination is sent over an RS-232 link to a remote location where it is compared (perhaps by the PC controlling the rest of your house?) to the correct combination. If it matches,

# FIRMWARE FURNACE

*Ed Nisley*

the central computer activates the electric lock. Indeed, you can vary the combination by time of day or use it to track which of several authorized people gained access.

Why not have the combination(s) stored in the 8031 so it can activate an output bit that drives the lock? Think about it: all a perpetrator need do is smash the box, rip out the guts, find the wire that obviously goes to the lock (because it has a relay on the CPU end!), and short the wires to the power supply.

Age and treachery will always overcome youth and innocence.

With that in mind, I've packaged the guts in a simple plastic case that is not suited for high-security outdoor use. I'll leave it to you to armor-plate and weatherproof as needed for your application. There is one unused I/O bit, so if you really want to add an unlock function, have at it. Just don't call me afterwards.. ..

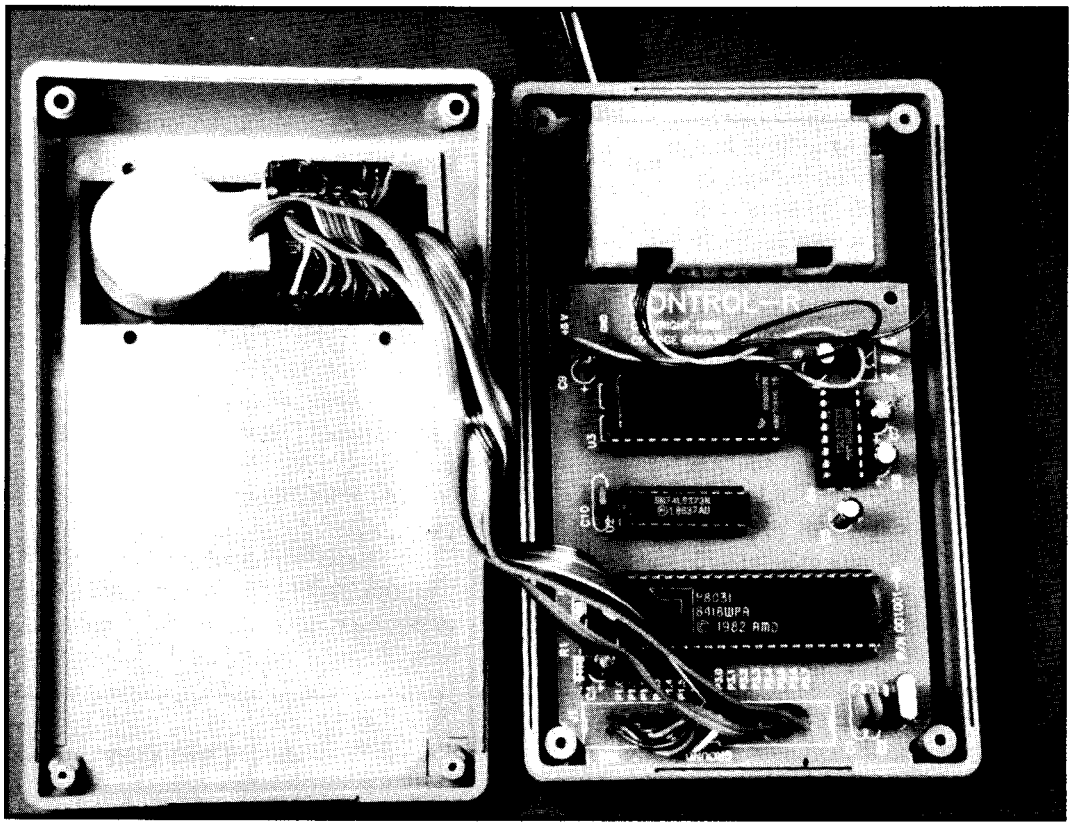


**Photo 1** — *The computer, rotary encoder, and display fit nicely in an off-the-shelf plastic box.*

## SENSING ROTATION

A push button is the archetypal digital input device because firmware can make two buttons produce "up" and "down" counting sequences. However, some situations call for a rotary knob. When you're peering into a 'scope while tweaking an output to perfection, the time-tested user interface is a knob: clockwise for more, counterclockwise for less, a fast twist for a bunch, a slow turn for an itsy, and hold still while you think. You don't have to wonder which button you're pressing and how long to hold it down before it auto-repeats on you....

While you can read a standard analog potentiometer with an analog-to-digital converter, a better solution is an all-digital, panel-mounted rotary encoder with TTL-compatible outputs. The processor interface requires just two bits: one generates an interrupt as the knob turns while the other indicates rotation direction. What could be easier?

The most common type of rotary encoder uses a masked wheel and a pair of photodetectors as shown in Figure 1. As you twist the knob, the wheel turns and the photodetectors report whether they "see" their respective LEDs through the mask or not. The mask shown in the figure produces only two pulses per revolution per detector, but real-world units will give you a hundred to several thousand pulses.

Figure 2 holds the key to using a rotary encoder. The photodetector outputs, commonly called Phase A and Phase B, are in "quadrature," which is a fancy way of saying Phase B has a 90" phase shift relative to Phase A. The sign of the phase shift depends on the rotation direction: -90" for clockwise, +90° for counterclockwise.

However, actually figuring out the direction doesn't involve trigonometry, phase angles, or complex calculations. Simply monitor Phase A for a l-to-0 logic transition; if Phase B is high immediately after the transition, then the wheel is turning clockwise. If Phase B is low, it's going counterclockwise.

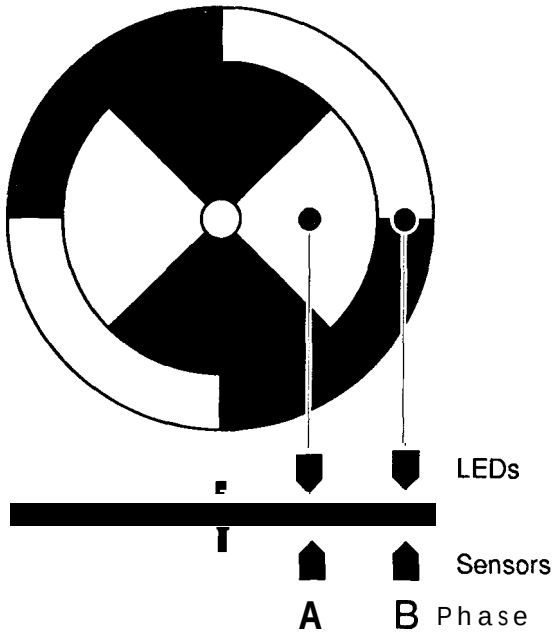



**Figure 1** —*This rotary encoder produces two pulses per revolution per detector.*

The most practical way to monitor the encoder is to write an interrupt routine triggered by Phase A. When an interrupt occurs you know that Phase A has just gone low, so reading Phase B gives an immediate indication of the direction. Each interrupt represents 1 /N-th rotation, where N is the number of pulses produced per full turn. A simple signed variable can hold the number of counts; the interrupt routine increments the variable for each clockwise event and decrements it for each counterclockwise event. A count of +N means the wheel has gone one full turn clockwise, -N means the same but in the other direction.

The connections for a Clarostat Model 600-128 rotary encoder are shown in Figure 3: Phase A goes to the INTO input on pin 12 and Phase B goes to pin 15. The encoder draws about 30 mA from the +5 V supply for its LEDs, photodetectors, and buffers. As you might guess from the model number, the 600-128 produces 128 counts per revolution and gives about 2.8' per count.

Incidentally, the patterns on the encoder wheel can be arranged in two ways. If you shift the Phase B pattern by 90" it will move the output signal by 180' (think about it) and reverse the sense of rotation. The Clarostat encoders use this pattern, so expect to flip your code no matter how careful you are.

Listing 1 shows the Micro-C code needed to update a 16-bit variable on each Phase A interrupt. Micro-C does not support the "interrupt" keyword provided by some other C compilers, but the in-line assembler code needed to save and restore the CPU registers is quite simple.

Micro-C can't handle a hardware bit directly, so the `TESTBIT` macro uses a few assembler instructions to copy the Phase B input bit into a C variable. `TESTBIT` ensures the variable will be 0 if the bit is low and 1 if the bit is high; although this verification is not strictly necessary, it makes tests against 1 meaningful.

Apart from those instructions, the code needed to decipher the encoder's state is a single `if` statement. That's simple enough.

Also, you can use encoder interface ICs that handle all the details without interrupting the CPU. These chips would be appropriate for high-performance applications that require extreme precision and rapid rotations. For monitoring a knob on a panel, firmware alone will suffice! Jeff tells me he's collecting information for the definitive encoder tutorial but won't commit to a date-stay tuned to his "From the Bench" column for more details.

With input in hand, we can look at the output display.

## CLASSIC CHARACTERS

Although LCD panels are probably the best overall display for small embedded systems, there are some situations where they are not appro-

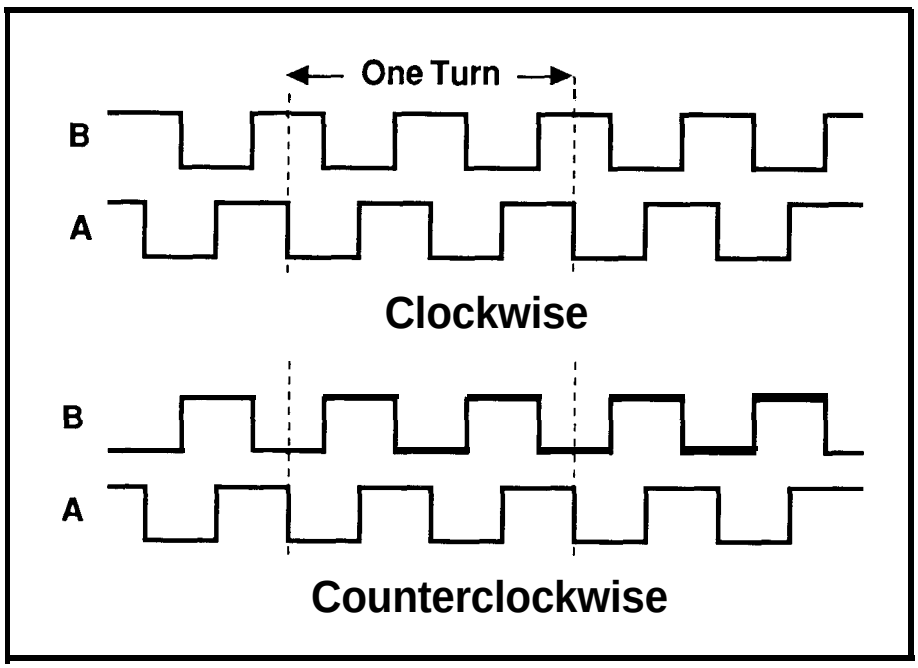



**Figure 2-Photodetector** *outputs nave a 9U° pnase shift that depends on rotation direction.*

priate. If you need high contrast, good night visibility, or just a tiny package, LEDs are hard to beat. The only catch is raw LED characters tend to use either lots of cheap components or a few expensive ones.

The display for this project is an oldie that solves the component count and cost problems: the Litronix DL-1416 4-character X-segment alphanumeric display. My data sheet is dated July 1979, which should give you some idea of when these things first came out.

Back when TV sets were small and dollars large, you could get a magnifying lens for that "big screen" effect. The DL-1416 has teeny-tiny LED segments and a big lens bubble over each character. You need to be directly in line with the optical axis to get a good view, but for a lock that's not a disadvantage at all: how many people do you want scrutinizing your combination?

The DL-1416 handles all the LED multiplexing and refreshing without an external clock input. It is not a low-power device: the data sheet specifies 35–100-mA drain from the +5-V supply. The interface is quite simple, with two address bits selecting one of four characters, six data lines for the ASCII character, and a write strobe to load the data. A chip enable simplifies multichip displays and a unique cursor function.

The DL-1416 timings are lethargic by contemporary standards. The address and data setups are 1000 ns and the strobes must be at least 500 ns wide. The simplest 8031 interface uses I/O port bits to drive the control lines, although other CPUs allow wait states to match the bus to the display.

Figure 3 shows the connections required to bring the DL-1416 to life. The -CE (Chip Enable) input is grounded because I only needed one unit, so the device is always enabled. Power goes to pin 18, ground to pin 19, and the remaining pins are not used.

Although the DL-1416 uses the ASCII character set, it includes only codes 20 hex (space) through 5F (underscore). This range includes the common punctuation symbols, numeric digits, and upper-case letters. The chip does not convert lower-case letters and does not include some useful symbols, but on the whole it beats building discrete LEDs.

## THE REST OF THE HARDWARE

Apart from the obligatory 8031 CPU, address latch, and EPROM, the only other chip is a MAX232, which converts serial port signals between RS-232 and TTL voltage levels.

As you can see in Photo 1, I used a Cottage Resources Control-R board for this project. While suitable for only the smallest projects because it has a 2764 EPROM and no provision for external RAM, it works well when the job fits. I had to nibble off only two corners to shoehorn it in the enclosure.

The Control-R does have an anomaly that can cause considerable headscratching. I routinely use an EPROM emulator instead of burning EPROMs, so I was surprised when neither of my emulators worked on the Control-R board. EPROM pin 26 turned out to be tied to 8031 pin 26, although it's not shown on their schematic. If the board were to use a 27256, pin 26 would be address bit 13 and the connection would make sense. However, the board uses a 2764, which shows pin 26 labeled as "no connection." The processor's Al3 was driving the pin to a state my emulators weren't expecting. Also, EPROM pin 1 is tied high, not low as shown in their schematic.

The catch is both of my EPROM emulators require all unused pins to be either disconnected or pulled up. After I tied pin 13 high, both emulators (and the EPROM!) worked correctly.

Moral of the story: take nothing for granted!

## INTRODUCING MICRO-C

With so little hardware, the firmware must obviously provide the majority of the lock's functions. Before diving into the code, I should explain a little about Micro-C so you know what to expect.


EXCLUSIVE
HOME AUTOMATION
GIFTS
FOR THE HOLIDAYS

AUTOMATION WEAR
* Limited Edition
* Top Quality
* Perfect Holiday Gift
* Let the world know you love home automation!




TURNS ME ON!

POLO SHIRTS
Handsome polo shirts by famous label OUTERBANKS, makers of premium clothing and golfwear. Home automation message beautifully silk screened on left chest.

SWEATSHIRTS
Top quality Hanes grey pullover Sweatshirts bear full-size home automation message.

LIMITED EDITION! ORDER TODAY!

ORDER FORM
Photocopies accepted

MAIL TO:
Home Control Concepts
9353-C Activity Rd.
San Diego, CA 92126

| | Specify Number of Sizes S M L XL | Unit Price | Total Price |
|---|---|---|---|
| Polo Shirt | | $29.95 | |
| Sweatshirt | | 29.95 | |
| Shipping & Handling | | | 5.75 |
| CA Residents 7.25% Sales Tax | | | |
| TOTAL | | | |

Name _____
Address _____
City _____ State ____ Zip ____
☐ VISA ☐ MasterCard ☐ Check or M.O. Enclosed
Card No. _____ Exp ____

Reader Service #149

The Micro-C compiler is a front-end translator that produces intermediate code. A CPU-specific code generator translates that code into assembly language statements, which then go through the appropriate cross-assembler to produce a HEX (or for PCs, EXE) file. The compiler and assemblers are available for a bewildering variety of CPUs: 80x86, 8080/Z80, 8051/31, 6811, 6809.. .essentially all the popular micros.

The C dialect is "K&R" rather than "ANSI," so much of what you read about C nowadays does not apply: no function prototyping, no argument checking, no single-line comments, and so on. There are some nonstandard extensions and quirky idioms ("register" instead of "near"), but the code is at least nominally portable to other compilers. It is entirely usable and a generation of C programmers got along well enough without the new ANSI standard features.

Of course, that same generation is responsible for C's reputation as a write-only language used to create programs full of bizarre tricks and obscure bugs. You get out what you put in: be neat, be discreet, and stay OK; be tricky and get dead.

The 8051 C compiler supports TINY, SMALL, COMPACT, MEDIUM, and LARGE **memory** models. You can use in-line assembler code for speed when it counts. The start-up code includes a tiny debugging kernel that you can use (I haven't, yet) to get your code working directly on the target hardware.

Dunfield also has an 8051 CPU simulator that shows the current CPU state, disassembles the instructions, and dumps Internal and External RAM. It can connect to a debugging kernel on the 8031 (through the serial port) to report and control the state of I/O ports. The current version works at the assembly language level and does not include symbolic information, but he is considering adding that feature and creating a C source level debugger as well.

The full-up Micro-C package comes with source code for everything: compilers, assemblers, run-time libraries, simulator, utilities, you name it. The 8086 compiler can rebuild itself, as well as the other programs, using the MAKE files included to simplify the process (you get the MAKE program and source, too). There are dozens of sample programs.

To put this compiler in perspective, everything described above will set you back slightly over $100 including shipping.

Obviously, we are talking obsession here. Dunfield's code is good, tight work that I'd be proud to have written. His style is somewhat idiosyncratic, as you would expect, but he knows what he's doing and can explain (or at least justify) his design decisions without retreating into mysticism. Once again, his products show that the best PC programs come from one person in pursuit of a vision.

And I have yet to hit a compiler bug. Enough said?

## MICRO-C AT WORK

As you saw in Listing 1, the Micro-C language requires some help to handle the (admittedly bizarre) 8051 architecture. The LED output routines provide a better look at what's needed to write hardware-specific code with a general-purpose language.

Remember, Micro-C was designed as a portable compiler to produce code for a variety of machines. The compiler is not the place for 8051-specific features and quirks; those be-

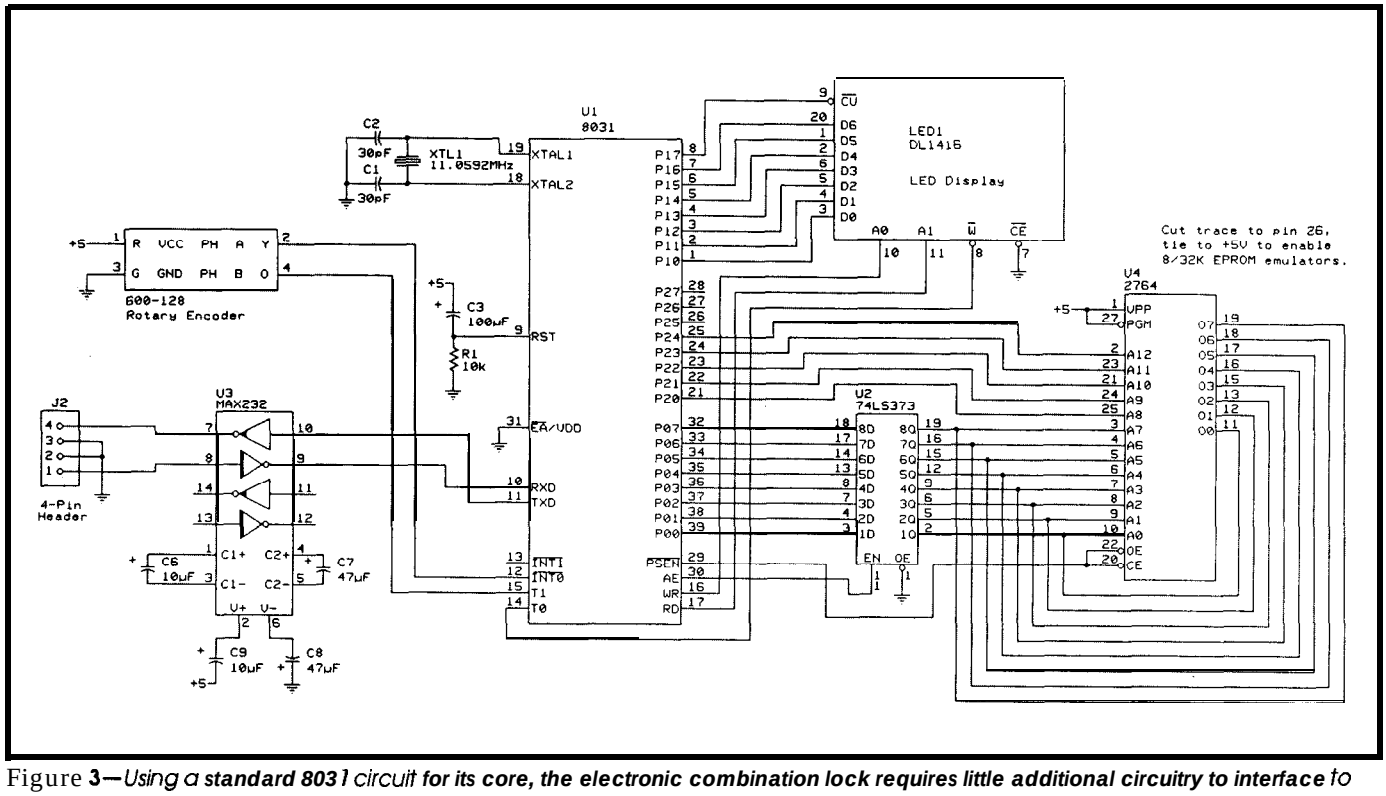



Figure 3—*Using a standard 8031 circuit for its core, the electronic combination lock requires little additional circuitry to interface to the rotary encoder and LED* display.

long either in the run-time library as callable functions or embedded in the code as in-line assembler sequences. You can buy tailored compilers, but they sacrifice portability for access to the 8051 market.

That's what tradeoffs are all about.

In any event, the code required to adapt Micro-C to the 8051 is not particularly burdensome. The TESTBIT macro in Listing 1is an example: a few instructions to set up a C variable from an 8051 hardware bit. Yes, It Would Be Nice if Micro-C would allow hardware bits in conditional tests, but a few instructions is not a big price to pay for generality.

The SETBIT and CLRBIT macros are #defined to insert a single assembler line into the C code: SETBIT translates into SETB and CLRBIT becomes a CLR instruction. Even a tailored language would be hard-pressed to do better than that!

WriteLED () starts with a loop that updates each of the four display characters. The DL-1416 puts character 0 on the right, but it is most natural for C strings to start with charac-

```
/*-------------------------------------------------------------*/
/* Global variables and bit definitions                        */

#define PHASEB P3.3      /* rotary encoder B phase input bit */
NEAR int  EncCounter;    /* raw encoder counts               */
NEAR BYTE PhB;           /* temp variable for bit testing    */

/*-------------------------------------------------------------*/
/* Bit test function                                           */
/* MC can't do an 'if'  directly on an 8051 bit variable       */

#define TESTBIT(byte,bit) \
#asm                  \
   CLR A              \
   MOV C,bit          \
   MOV A.0,C          \
   MOV byte,A         \
#endasm


/*-------------------------------------------------------------*/
/* External Interrupt 0 handler                                */
/* This is triggered by Phase A of rotary encoder going low    */

ExtInt0() {


/*--- save regs                                               */

#asm

VExtInt0 EQU  *
   PUSH  A
   PUSH  PSW
   PUSH  B
tendasm

/*--- decide which way we're turning                          */

   TESTBIT(PhB,PHASEB);
   if (PhB) {
      --EncCounter;
   }
   else {
      ++EncCounter;
   }

/*--- restore reqs and return...                              */

#asm
   POP   B
   POP   PSW
   POP   A
   RET1                          /* preempt normal return    */
tendasm
```

**listing 1 -Each** *encoder Phase A pulse produces an interrupt on the INT0 line.* **This interrupt handler checks the** *PHASEB input to determine which way the encoder wheel is turning and adjusts the EncCounter variable accordingly.*


HOME CONTROL CONCEPTS PRESENTS...

HACKER HEAVEN

X-10 DEVELOPMENT KIT

Use to develop your own PC-based "smart" home automation system! Monitor status of home's lights & appliances and make intelligent decisions based on their on/off status. Develop a home control system with IF-THEN logic, even 1-button macros! Add Stanley motion detectors to give system input of room presence. Development software is interrupt based (does not use polling!) and includes compiled library routines and sample C-language source code.

Use with PC *to Infrared Interface* to develop a system which combines home automation and IR control; any X-10 controller can control infrared! Add *Voice Master Key*, for voice control of the home! Use X-10's *Sundowner* to give dusk/dawn input to your system. Add PC voice mail for X-10 control of your home from any phone i" the world!
Requires IBM or compatible computer with parallel port. Includes TW523 module, adapter, interface cable, development software, demo program & technical info/data. HCC-523K ONLY $69!

WIRELESS RF LINK

Manufactured by Linear. this low cost RF hnk is ideal for wireless control of your own projects, your home and car alarm, and even X-10 modules (with addition of X-IO Burglar Alarm Interface)! Set security code on transmitter and receiver, apply power to receiver board, and you're ready for wireless control!

TRANSMITTER: Tiny keychain transmitter is approx. half the height of a matchbox! Transmitter has two buttons corresponding to channels 1 and 2. Includes two Lithium batteries.

RECEIVER: Board level receiver measures approx 3" square! Requires power supply of 8 to 24 VDC or 12 to 18 VAC. Two outputs (channels 1 and 2) can each switch up to 300 mA @ 18 VDC maximum to ground. Directly activate relays, drive bulbs, mae.

Set includes transmitter. receiver, data. HCC-RF1 ONLY $39!

PC to IR INTERFACE KIT

This kit is great la development of your own infrared home control system! Allows your PC to "push buttons" on remote control! Combine PC based home automation with infrared control of your TV (volume, channel. et), stereo, VCR, and more! Add whole-house IR repeater such as X-10's *Powermid.* Jse with Covox Voice Master *Key for* voice control of your entertainment system! Combine with *X-10 Development Kit* to allow any X-10 controller to control your infrared deuces! Use with voice mail system for remote control of IR from any telephone n he world! Possibilities are limitless!

Remote control links to PC's serial port. Use the SendIR program to transmit infrared signals (e.g. The dos command SendIR TV MUTE will mute the tv!). Write your software around SendIR or develop from scratch using sample source code. Complete with Interface, cable, development software, sample C-language source code, technical info/data and documentation. Requires remote control *(One-For-Alla 1-Control)* and IBM or compatible computer with serial port. HCC-PCIR ONLY $79!

USEFUL PLANS

HCC-PLN1 Add local dimming (from the switch itself) to your X-10 Wall Switch! Push on / push off; hold in button for dim/bright. Capability is designed into the switch (br Leviton's specs) Simple to modify; just cut two parts and adda jumper tire. For informational purposes only. Complete plans $6.

HCC-PLNZ Control your Hi-Fi's volume with X-10 dim/bright buttons! For informational purposes only. Complete plans $6

HCC-PLN3 Cable television secrets; watching pay channels. For informational purposes only. $9.

HCC-PLN4 Modify X-10 modules for momentary on or off. Or, modify mini-controllers for momentary on or off control; just add a jumper Hire! For informational purposes only. Plans $6.

HCC-PLANS All 4 plans for only $19! (Reg. $27)
HACKERS BONUS Order 2 items from this ad, get FREE plans set!

HOME CONTROL CONCEPTS
1-619-693-8887
ORDERS 1-800-CONTROL
See Our Full-Page Ad In This Issue

Reader Service #150

ter 0 on the left (you think of `LedChars[0]` as coming before `LedChars[1]`). The six lines of assembler invert the `Index` variable to change the character order and transfer the value to the output bits driving the DL-1416 address pins.

You could use this Micro-C statement to set the DL-1416 address bits within the loop:

```
P3=(P3&0x3F)|(~Index<<6),
```

which might (ideally!) compile to something like:

```
MOV   A,P3
ANL   A,#$3F
MOV   B,A
MOV   A,Index
CPL   A
RR    A
RR    A
ANL   A,#$C0
ORL   A,B
MOV   P3,A
```

But immediately after that C statement executes, the rotary encoder stops working. The reason is the P3 bits picked up on the right side of the statement (by `MOV A,P3`) include the current encoder output bits. If either of those bits is zero, the value assigned to P3 on the left (by `MOV P3,A`) sets the corresponding output latch bit to zero.

Now, we all know the input value read from an 8051 CPU port is the logical AND of the output latch and the pin state, right? So there should be no surprise that a zero latch forces the input to look like zero. Hmmm?

Even if you use a tailored language you must know exactly how the hardware works to get the correct result. I'd be slightly embarrassed to admit how often I've flubbed this one, so we shall discuss it no further. However, the error is much, much harder to find when a high-level language makes I/O operations look just like memory operations; after all, if an operation works on variables, remembering it won't work on ports is hard.

The bottom line is that the relatively few things Micro-C can't handle directly can be simulated by C (or

```
/*-------------------------------------------------------------*/
/* Global variables and bit definitions                        */

#define NUMCHARS 4          /* number of LED characters        */

#define LEDWR  P3.4         /* LED -WR write strobe            */
#define LEDCU  P1.7         /* LED -CU cursor enable           */
#define LEDA0  P3.6         /* LED address bit 0               */
#define LEDA1  P3.7         /* LED address bit 1               */
#define LEDDATA   P1        /* LED character data              */

#define BLINKCODE     ' '   /* "off" character during blinking */
#define CURSORCODE    ' '   /* background when cursor goes away*/

/*-------------------------------------------------------------*/
/* Bit set and clears                                          */

#define SETBIT(bit) ; \
#asm \
   SETB  bit  \
#endasm

#define CLRBIT(bit) ; \
#asm \
   CLR   bit  \
#endasm

/*-------------------------------------------------------------*/
/* Write all digits to the LED display                         */
/* We invert the order of the characters to make LedChars      */
/*  readable and easy                                          */

WriteLED() {

   for (Index = 0;
        Index < NUMCHARS;
        ++Index) {
#asm
      MOV   A,Index
      CPL   A                        /* swap left<->right     */
      MOV   C,A.0                    /* set up address bits   */
      MOV   LEDA0,C
      MOV   C,A.1
      MOV   LEDA1,C
#endasm

      TempChar = LedChars[Index];    /* pick char from array  */

      TESTBIT(TestBits,FlagBlinkEn);
      if (TestBits) {
         TESTBIT(TestBits,FlagBlink);
         if (TestBits) {
            if (Index != CursorIndex) {
               TempChar = BLINKCODE;
            }
         }
      }
      LEDDATA = TempChar;
      SETBIT(LEDCU);              /* turn off cursor strobe    */

      CLRBIT(LEDWR);              /* write the data bits       */
      SETBIT(LEDWR);

   }

/*--- set up cursor character overlay                         */
/*    the cursor is always written and goes OFF by default     */

   TempByte = 0;                 /* assume cursor should go off */
   TESTBIT(TestBits,FlagCursorEn);
   if (TestBits) {
      if (CursorIndex < NUMCHARS) {
         TESTBIT(TestBits,FlagCursor);
         if (TestBits) {
            TempByte=0x08 >> CursorIndex;/* put cursor on char */
         }
      }
   }
   LEDDATA = TempByte;           /* Cursor Enable bit is 0 = ON */

   CLRBIT(LEDWR);                /* write the cursor bits     */
   SETBIT(LEDWR);
}
```

**Listing 2**—*The DL-1416 LED display is controlled entirely through CPU I/O port bits. The firmware must set up the data and toggle the control lines to write characters and position the cursor.*

assembler) functions or in-line assembler code (perhaps disguised by a `#define` to look like a function).

The remainder of the C code for the combination lock looks just like C code you'd find anywhere else. That's the good thing about C; most of the code in any project has nothing to do with the 8051 itself, so using a higher-level language to get the most done with the least effort makes sense. The digital lock's complete source code is available on the Circuit Cellar BBS as usual. *[Editor's Note: Software for this article is available from the Circuit Cellar BBS and on Software On Disk #24. See page 105 for downloading and ordering information.]*

## IS MICRO-C FOR REAL?

Despite my admiration for Dunfield's work, Micro-C is probably not the ideal C for very complex projects. I've gotten used to ANSI C features and the error checking built into current compilers. In addition, high-end compilers support full C source code debugging, which greatly reduces the time required to get a project running. When you're doing this stuff for a living, the return on investment can be compelling.

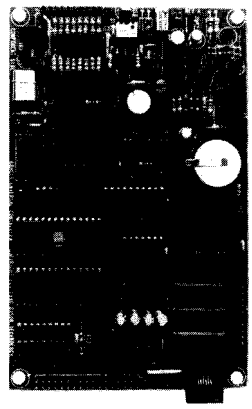
With that in mind, I plan to use Micro-C for many upcoming Firmware Furnace projects. The compiler produces code for Motorola micros as well as the Intel family, but I don't expect to write much portable firmware because the bit blasting doesn't convert nicely. Stay locked in for some interesting stuff.. .and suggestions for topics are always welcome.

If the price of entry has kept you out of the microcontroller C market, you have no further excuses. Micro-C is as good as it gets!

## CREDITS AND SOURCES

David Penrose's "Electronic Combination Lock" entry in the Second Circuit Cellar INK Design Contest inspired the topic. He captured Third Place in the Cost Effective category and followed up with a chapter in *The Circuit Cellar Project File, Volume I.* His lock used a 8748 CPU


# Ten microwatts.



With its automatic **power cycling,** that's the average power draw of the nanoLINK Controller/Peripheral in a typical application. No extras--it's all built-in and can be enabled with one instruction. Run your application for over two years on a 9V battery, or forever on a thumb-sized solar cell.

And because the nanoLINK Controller/Peripheral was designed to function as a PC peri heral--no programming required--OR as a stan8alone controller, you can use it on the bench, in the field, or both. 8 A/D inputs, 32 I/O lines, 1A power & PWM outputs, and much more.

A full-function BIOS is included--no device drivers to write. A typical data-logging application requires less than a dozen lines of code. And an interactive editor & macro assembler is included--no expensive cross development tools to buy.

And the price? $395, quantity l--call today.

Less money, less power, less time, less work.

**To place an order, or for more information, call:**

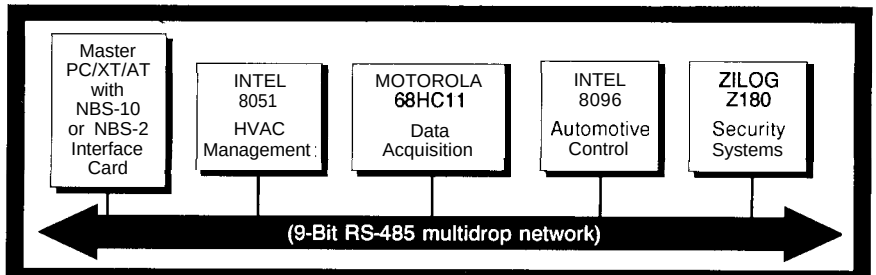
**1-800-GET-DATA (438-3282)**

**1-602-996-0255/fax**

**nano| Power | Devices**

A division of AirDigital Corporation

Reader Service #103




# Embedded Controller Networks
# Low Cost - High Performance

Cimetrics Technology's 9-Bit Solution networks popular 8- and 16-Bit microcontrollers by taking advantage of the built-in 9-Bit multiprocessor modes.

## 9 Benefits of the 9-Bit Solution

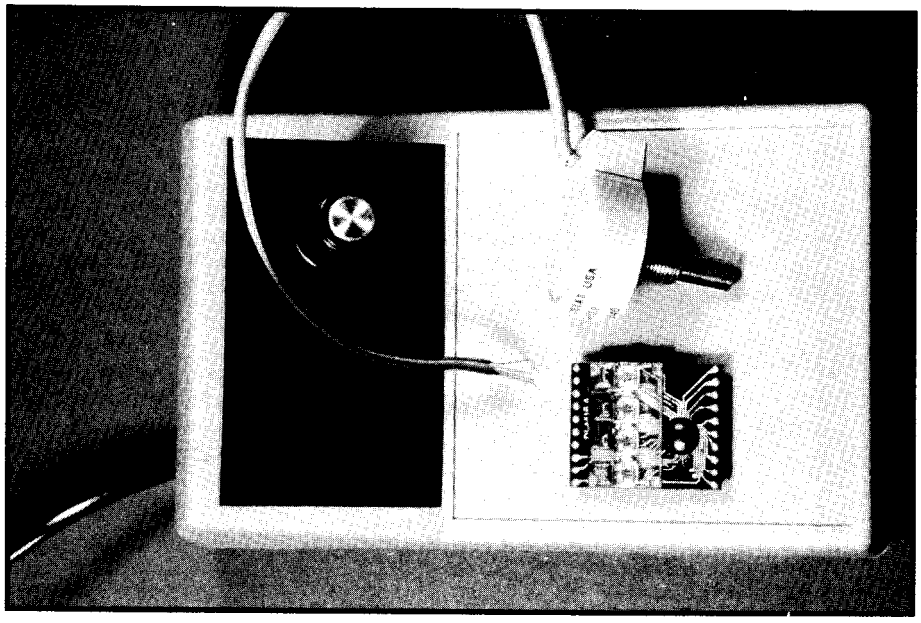
- Lowest cost embedded controller networking alternative
- Compatible with your micro-controllers
- Multi-drop Master/Slave RS-485 network
- 250 nodes per network
- High speed (62.5K baud) with low network overhead
- High reliability protocol includes CRC checking
- Low resource requirements no external RAM required
- Simple to use software toolkit

Applications in include data acquisition and distributed control

Cimetrics Technology's Y-Bit Solution includes the NBS-2 ($165) or NBS-10 ($249) PC/XT/AT network interface, the NSP ($750) or Tiny-NSP ($199) network protocol software toolkit, and network monitor software ($225).

**Cimetrics Technology**
120 West State Street
Ithaca, New York 14850
(607) 273-5715 / (607) 273-5712 FAX

Reader Service #121



and 7-segment LEDs controlled by a clever assembly language program, so our projects don't have much in common apart from the rotary encoders. Indeed, you might want to compare how our programs solve the same problems in different ways.

A demo version of Dave Dunfield's Micro-C is available on the Circuit Cellar BBS, but it produces Intel 8086 code for the IBM PC and is not directly useful for 8051 projects. He has bundled the 8051 Micro-C compiler, cross-assembler, simulator, and a bunch of handy utilities into an "8051 Developers Special" for $99.95 plus $4/$6 for surface/air shipping to U.S. or Canadian addresses.

He will send a catalog of his products and prices if you send a stamped self-addressed envelope. If you can't wait to order, make your check payable to Dave Dunfield at: Dunfield Development Systems, P.O. Box 31044, Nepean, Ontario (Canada), K2B8S8. Tell him I sent you...

The Control-R board is made by Cottage Resources; see their ad in the pages of this magazine for details.

The rotary encoders and LED displays are available from your usual electronics distributors. They also appear regularly in mail-order surplus outlet ads, so keep your eyes open for good deals. My DL-1416s came from my junk box, so I haven't actually placed an order for them in a long time.

*Ed Nisley is a Registered Professional Engineer and a member of the Circuit Cellar INK engineering staff. He specializes in finding innovative solutions to demanding and unusual technical problems.*

## IRS

416 Very Useful
417 Moderately Useful
418 Not Useful


# ERAC CO.

SINCE 1974

**P.O. BOX** 1108, **POWAY CA.92074-1108**
VOICE (619) 6798360 FAX (619) 6796670
VISA or **Master Card (add 3.3%), Check or Money order. no CODs, Calif. orders add 8.25 %.**

**Motherboards**

**486** 20SX/33DX/64K Cache/Baby $93711137
486 20SX/25DX/64K Cache/Baby $899/1036
486-33DX/64 Cache, Full size 1014
386-40 DX,64KCache,16M,0K,AMI,baby 475
386-33 DX,64KCache,8M,0K,AMI,baby 445
286-12, C&T/Half size,4M,sipps,0K,AMI 89

**Backplane Boards**

XT backplane,blank 9 or 15 slot 1"$7.50/9.00
AT Backplane 8slot, KB & PS Conn. 35
386-25 DX,8M,0K,AMI,2 ser.,par. 395
386-16 SX.8M,0K,AMI,2 ser.,par.,IDE/FI 335
286-6 Uses ext. mem. brd 35
Mem. board, AT,640K uses 64K dram 35

**Cases**

Full Tower 5-51/4",2-3.5"Hd,230W Ps $169
Tiny Tower 2-51/4",2-3.5",3.5"Hd,200W 91
Small AT 3-51/4",1-3.5", 200W 115

**Miscellaneous**

Pioneer DRM600, 6 Disk changer(demo)$945
CompaqLTE386s20,2M,60M,VGA,mdm 2500
Cyrix Fastmath Co-proc. 80D387-33 199
Intel 80C287A-12 Co-proc. For laptops 91
AT I/O 2serial(com1-4),parr.,game 20
2400/1200 Baud Modem, int., adv.err.corr. 56
AT PS, Full size 270 W, for tower cases 65
Elgar FS350 UPS with Failsafe software 195

**Instrumentation**

Fluke 2240C 60 Channel Datalogger,40 ch low level,10ch standard,8 ch RTD, Instr. Man. $995
HP 3998PInstrumentation Tape Rec. 8-Ch 2295
OCTAPAC OP90 New 220
OCTAPAC OP80 New 220
Visipac 524Programable Load Cell Monitor 299
*SPECIAL*Industrial Temperature Controllers
These PID controllers are compact and have a LED display.Inputs are either C,mV or RTD.
DC7000 PID Single Loop,PID with automatic tuning, output limiter, H/L Dev. Alarms,LED Bargraph. Output options available 85
DC7100 Like the DC7000 plus 7-segment Ramp and Soak program capability 120
DC7200 Auto-tune,rem/multi setpoint,full PID130
DIGILARM2000,24ch.TC,mv,RTD4-20ma..Output:2 sep. alarms, 4-20ma,deg C/F 495
DORIC-PAC's many models, Call for info.
Beckman460 (Doric 700, Omega DP-85) Digital Process Indicators, uP Based,multi chan.,versital, expandable,Menu Driven, list 549.new 275
Beckman470 (Doric 710, Omega DP-86) similar to above but 5 digit res., list 649...new 310
Action instr. TC signal-conditioning, IsolatedTC-Xmit, linearized, 12ch + card-cage, Deg C/F, V/I output,Var. gain out. 495
Beckman Instr.,DM15b, DMM (used)tested 25

**Chart Recorders**

Yokogawa 4156 100mm 6Ch Recorder. These new hybrid recorders are loaded with features,6 colors,chart speed(l-1500 mm/hr) indep.prog. input types ( Mix thermocouple types and DC voltranges), Prog. Full-scale ranges, Back-lit digitaldisplay and Analog Bar Graph,Linear Scaling Functions, Delta T Recording, Batt.backed program and Real-Time Clock, Print-out of:Tags Program, Alarms, Time, Date.Current model.
List $1895 While they last $735
Honeywell AR100 7-day 10" Circ.Rec/Contr. (J,T,K,mV,V,RTD) 250-charts 250
HP9872C Plotter 11X17, 8 pen, IEEE488, HPGL 750
OMEGA RD145,100mm,100/500C, J, charts 175
OMEGA 5830 Therm. thermometer 5Ch. for 400/700 series probes, min/max limit, avg, scan, diff. 275

**Transducers**

Yellow Springs Precision Thermistors with 10' of vinyl lead and phone plug, data sheet and a Certificate of Traceability, NEW.
YSI402 Small Flex Vinyl Sheath &Tip $10
YS1405 Air/Gas stream Probe, stainless 15
YSI703 Ss liquid immersion probe 12

Reader Service #138

# Electronic Identification

FROM THE BENCH

Jeff Bachiochi

Rockville General Hospital, September 20, 2005.

"Congratulations Mr. B, you have a new son. We've just finished implanting a PIN microchip under his thumbnail, and his Personal Identification Number is #USA-CT-9-20-2005-01 36," Doc announced. "You know, ever since the World Council mandated PIN implants at birth, I've felt more like a technician and less like a doctor. It won't be too long before every living being on this planet can be identified and their whereabouts pinpointed in just a matter of seconds," he added proudly. 'Isn't technology wonderful?"

"It certainly has made our lives easier, there's no doubt about it," I acknowledged. 'No need to carry ID or credit cards any more, this little implant does it all."

## FACT OR FICTION

Is this view of the future far from the truth? Today we fanatically organize, alphabetize, and enumerate our lives. We have numbers for everything: phone numbers, house numbers, zip codes, even time and date codes. The decimal system is no longer adequate; we have to have binary, hex, octal, hexadecimal, Roman numerals, and so on.

Take your social security number for instance. Nearly every form you fill out or bill you receive has this identification on it, and you can't even open a checking or savings account without one. You and I are just numbers in the massive data bank of the IRS.

## HALT, WHO GOES THERE?

What is it that links our social security number to us? Fingerprints? Photographs? Magnetic strips or our signatures? How can these items lead to our identification? They all have a common thread. Differentiating aspects within each item, such as a fingerprint with a whorl versus a loop, a picture of a blond versus a redhead, or your credit card's one versus zero flux orientation, help separate one from the other. The total number of possibilities for an aspect make up a set. A 16-bit binary set like the address range of an 8031 has 65,536 possible IDs (or addresses). Our social security number has a very large set: 1,000,000,000 possible combinations. Your ATM machine input pad uses the same $10^4$ (10,000) set as your telephone, and security is based on the impracticality of trial-and-error guessing because of the large number of combinations.

A signature is adequate identification for our everyday transactions. However, signature recognition by computer is still in the Stone Age and not ready for the mass market, so we must still rely on more conventional means of identification. Present forms of identification may include your house key, credit card, safe combination, library card, or a password. Some of these forms are more secure than others. For example, the house key, which identifies the bearer as "one who may enter," can be copied at any hardware store.

A door key also uses a set of possibilities to specify a combination. Your house key's notches must match the pin lengths within your door's lock, or the key can't rotate the tumbler and remove the bolt from the doorjamb. Each notch is more than the binary value of notch/no notch because most keys can have notches of varying depths. A trinary or quadruple value can be associated with each notch. A key with eight notches of binary depth has a set

of possibilities totaling $2^8$ (256), and an eight-notched trinary depth key's set is $3^8$ (6561). Naturally, the greater the number of combinations, the more secure the set. So using a housekey for security is based on a combination set that is large enough to be impractical to break and because you can prevent it from being copied.

## HARDWARE AND SOFTWARE SECURITY

If you have an AT, you probably have a hardware lock on the front of the box. This lock enables and disables keyboard entry. Without the proper key no keyboard entry can be made and the system is safe. You may also have some variation of password protection to prevent

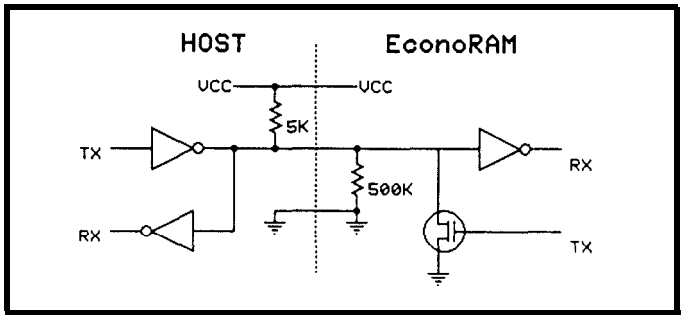



**Figure 1 — The *host interfaces at its 4.7k pull-up with the 500k pull-down of the EconoRAM.***

unwanted access either to your system or to certain file areas. Both hardware and software keys establish identification from their own fixed set of combinations and offer a level of security.

Hardware and software securities are oftentimes intertwined. Take a look at the ever controversial dongle. This piece of hardware sticks to your computer's I/O port and perks up under the right conditions to send back information to the system. The software that started the transaction then reads the data sent by the dongle and executes a conditional jump based on that data. This jump could be a friendly message informing you the dongle is not attached followed by a quick boot out of the program if the data is not as expected, otherwise normal execution is resumed. If the software is really nasty, it could calculate how the dongle would respond and adjust the base address of a jump. Then the data with the added offset would equal the next program line to be executed. Data that does not indicate a dongle would add an incorrect offset to the base address and jump to somewhere inconsistent with the normal program flow. This inconsistency could create anything from minor to major havoc. Code of this type is akin to intentional virus planting and as such would be a likely candidate for legal suit. Let's save the exploration of dongles for another day.

## ID IN A TO-92

Dallas Semiconductor released the product EconoRAM a couple of years ago. At the time the press release crossed my desk I read it and thought, "RAM in a TO-92 package, WOW!" But then I read further and discovered it was only 256 bits, which made the calisthenics needed to communicate hardly worthwhile. So into the file cabinet it went along with all the other orphan ideas looking for good homes.

Every orphan deserves a good home, and I think I've found one for EconoRAM. It can be used in the security of your system. First, at little background. EconoRAM is a low-cost, 256-bit memory device that has reduced its control, address, and data interface to a single pin. Include power and ground, and you have a grand total of three pins. Operating voltage is from 1.2 to 5.5 V at a current of only a few nanoamps. Data retention standby current is typically less than 1 nA.

OK, so how does this 256 x 1 fully static RAM work with only one lead? The host interface consists of a bidirectional bit with an open-drain driver and a pull-up of about 4.7k. EconoRAM has the same interface except it uses a 500k pull-down. This pull-down keeps the data pin low when the device is disconnected, but it allows the pin to be easily pulled high by the host when connected. If the host outputs a logic "1," EconoRAM can pull the data line to a logic "0" state because of this wired-OR connection. Figure 1 shows the interface necessary for communication with EconoRAM.

The host is responsible for initiating a read or write bit stream sequence. Each bit of the sequence is a time slot. The time slot begins on a logic "1" to logic "0" transition and lasts for a minimum of 60 µs. Three types of time slots are

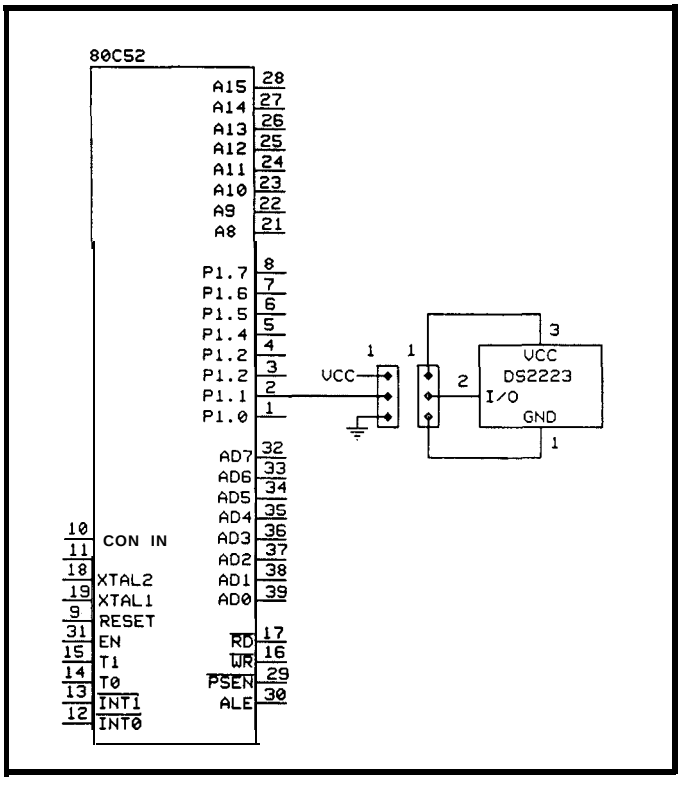



**Figure *2-If the power source for the EconoRAM is always from a battery, then the number of connections involved in the 80C52-to-DS2223 interface falls from three to two.***

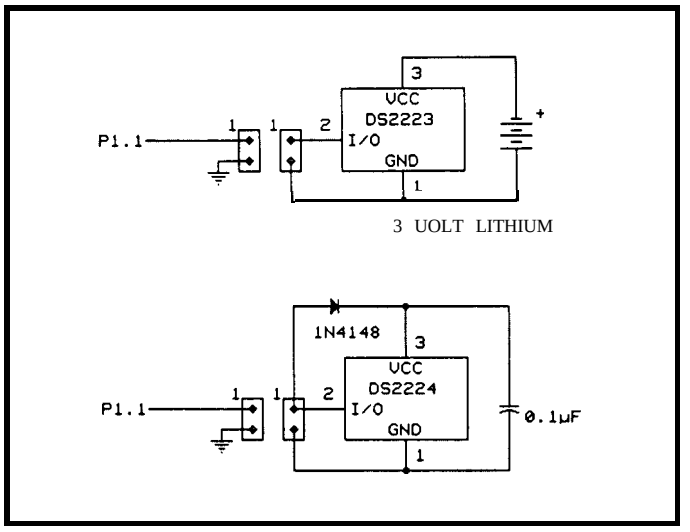



**Figure 3**—*The 80C52-to-DS2223* **interface shown with two** connections and powered by either a lithium battery or the data line.

possible. A "zero" time slot remains at logic "0" for a minimum of 60 µs. A "one" time slot returns to a logic "1" after a maximum of 15 µs at logic "0" and remains at logic "1" for the rest of the 60 µs. A "read" time slot must return to logic "1" after 1 µs to allow the EconoRAM to place its data on the line. The EconoRAM's data is valid for a

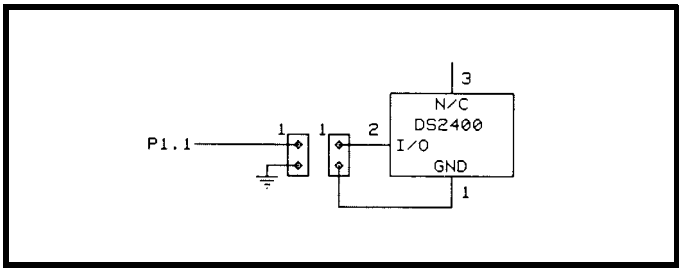



**Figure 4-A** data line can be used to power the DS2400 because the capacitor retains enough of a charge to power communications.

minimum of 15 µs and should be read by the host during this time. The data line will remain idle for the rest of the time slot.

Communication is initiated with a sync sequence of 264 "zero" time slots. This sequence increments the internal address pointer of EconoRAM to its maximum count, assuring synchronization to the command sequence that follows. The command sequence is a command word followed by 256 additional time slots. The command word is either F9H for writing 256 bits of data or F1H for reading 256 bits.

The host must issue eight "one"/"zero" time slots for the command word followed by 256 "one"/"zero" time slots to write data or 256 "read" time slots to read data. If the internal address counter is at its maximum count, reception of a command word will reset the counter and institute a new sequence. Therefore, be sure to always finish a complete read or write cycle to return the device to its known sync state and the address counter to maximum count.


TIMELINE INC.

liquid Crystal Displays
16 Characters x 1 line 3 for $25.00
40 Characters x 2 line 2 for $25.00
16 x 2 $12.00 ..... 32 x 4 $20.00
20x2 $12.00 ..... 4x2-55
24 Characters x 2 line LCD 12.00
5V power required • Built In C-MOS LCD driver & controller . Easy 'Microprocessor' interface • 98 ASCII character generator

4DAPTEC 4070A (RLL) or 4000A (MFM) SCSI Controller, your choice $60.00

SHARP-LASER DIODE
Part#: LT022MC $5.00
5mW at 780 n.m. single transverse mode

3 bit IRMA BOARD $99.00
Links 3270 mainframe systems to IBM PC

MAGNETIC CARD READER $25.00
Includes: • 20 character dot matrix display with full alpha-numeric capability • keypanel with full alpha numeric entry • separate 7.5 VDC/0.5 Amp power supply • standard telephone interface extension cord • lithium battery and flat-cone speaker.

FLOPPY DRIVES
360 K $45.00 1.2 Mb $55.00
720 K $40.00 1.44 Mb $60.00
5.25' bracket for 720K Of 1.44 Mb drives $7 extra

640 x 200 LCD $29.00
The unit is the display for the Toshiba T1100+ laptop computer. Driver board to generate text and graphics not included. This is a serial interface LCD. You should be very familiar with wiring when attempting to use this unit.
640 x 400 LCD $50.00 Seiko part #M402

380 WATT POWER SUPPLY $29.00
Non-enclosed • Output: +5V @ 50 A, +12V @ 6 A, -12V @ 1.5 A, Dimensions: 12" L x 5" W x 4"H

73 WATT SWITCHING POWER SUPPLY $19.00 or 2 for $30.00
(2) 4 pin power connectors attached • 115/230 Volt Dimensions: 8.5" L x 4.5" W x 2" H • Output: • +5V @ 2-9.75 A • +12V @ 0-1.5 A • -5V @ 0-0.4 A • -12V @ 0-0.5 A

40 WATT NON-ENCLOSED POWER SUPPLY 2 for $25.00
50 WATT NON-ENCLOSED POWER SUPPLY 2 for $25.00
60 WATT ENCLOSED POWER SUPPLY 2 for $25.00

NEGARGON LASER $399.00
30 Mw. (avg). Blue Bean only, Model 3030. Fully tested. Power supply available for $500.
HeNe Laser Tube 10Mw (max. output) $99.00

Optical Laser Array $499.00
Call for free information package

Magnavox EGA color monitor
$150.00
.39 dot pitch. Model #:9CM053. EGA card available

HARD DRIVES
10Mb Drive $50.00
(Half Height, ST506 Compatible 612 Cylinders, 2 heads)
20Mb Hard Drive $120.00
CDC 94166-141 $449.00
(115Mb, Full Height, ESDI interface)

Non-Enclosed TTL Monitors
Comes with pinout. 12V at 1.4 Amp input. Horizontal frequency 15 Khz. Ability to do 40 and 80 column.
5" Black & White....$49.00
7 inch..... $29.95
9 inch (Amber)..... $29.95

(80 I 16 line) Graphic & Alphanumeric
480 Dot x 128 Dot LIQUID CRYSTAL DISPLAY
15.00 each, or 2 for 20.00
Driver board available 60.00

101 Key XT Keyboard $24.00
85Key XT/AT Switchable Keybd $24.00

CONTROLLER CARD BLOWOUT!
XI hard drive controller
Controls 2 hard drives ..... $39.00
MFM controller-1/2 card-8 bit with cables $44.00
AT Hard Drive controller 16 bit
Part #WD1003-WAH - runs up to 12 Mhz only. $29.00

80387-s
These are - 16DX Mhz co-processors. Fully tested and guaranteed to work at the desired speeds below.
16Mhz — 115.00 20Mhz — 120.00
25Mhz — 125.00 33Mhz — 135.00

ProteonProNet-4p1347 Token Ring Board $129.00
16 bit • 4 Mbps • IEEE 802.2 and 802.5 compatible • twisted pair • interoperable with IBM Token Ring network

AMPRO LITTLE BOARD PC $79.00
256K RAM included, expandable to 768K on board (256x9 SIMMS) • Floppy Controller 360K/720K/1.2Mb/1.44Mb • 2 serial ports and 1 parallel port • Keyboard input 6 Speaker Output • Has SCSI controller socket • (requires optional pal & controller) • Board is the size of a 5.25" floppy drive

THE SPY IN THE SKY
1728 element CCD $19.00
4096 element CCD $29.00

1490 W. ARTESIA BLVD, GARDENA, CA 90247
Order desk only: USA: (800) 872-8878 CA: (800) 223-9977
L.A. & Technical Info: (213) 217-8912 • OEM INQUIRIES WELCOME
Fax (213) 532-6304 • 20% Restocking fee for returned orders.
Minimum Order: $29.00 Minimum shipping handling charge $5.00. We accept cashiers checks, MC or VISA. No personal checks or COD's. CA residents add 8.25% sales tax. We are not responsible for typographical errors. All merchandise subject to prior sale. Phone orders welcome. foreign orders require special handling. Prices subject to change without notice.

Reader Service #200

```
10 DBY(3EH)=6DH
20 PRINT
30   PRINT   "Hit."
40 PRINT "1 - SYNC"
50 PRINT "2 - READ"
60 PRINT "3 - WRITE"
70 PRINT "4 - PRESENCE"
80 PRINT "5 - END"
90 G=GET
100 G=GET
110 IF (G<31H.OR.G>35H) THEN GOT0 100
120 ON G-31H GOT0 130,150,210,280,510
130 CALL 6000H
140 GOT0 20
150 CALL 6003H
160 FOR X=4DH TO 6CH
170 PHO. DBY(X),
180 NEXT X
190 PRINT
200 GOT0 20
210 FOR X=4DH TO 6CH
220 PRINT "Value #",X-4DH," to be saved into
     register",
230 PHO. X, : INPUT V : IF (V<0.OR.V>255)
     THEN GOT0 220
240 DBY(X)=V
250 NEXT X
260 CALL 6006H
270 GOT0 20
280 CALL 6009H
290 FOR X=4EH TO 55H
300 PHO. DBY(X),
310 NEXT X
320 PRINT
330 IF DBY(4DH)=0 THEN PRINT "NO PRESENCE
     DETECTED" : GOT0 20
340 IF DBY(4EH)=01H THEN PRINT "FOUND DS2400"
     ELSE PRINT "NO RECOGNITION"
350 PRINT "CALCULATING CRC"
360 C=O
370 FOR B=4EH TO 54H
380 V=DBY(B)
390 FOR Z=0 TO 7
400 W=V.XOR.C
410 F=W.AND.1
420 W=C
430 IF (F=1) THEN W=W.XOR.18H
440 IF (F=1) THEN C=INT(W/2)+128 ELSE
     C=INT(W/2)
450 IF (V.AND.1)=1 THEN V=INT(V/2)+128 ELSE
     V=V/2
460 NEXT Z
470 NEXT B
480 IF C=DBY(55H) THEN PRINT "CRC OK" ELSE
     PRINT "BAD CRC"
490 PRINT
500 GOT0 20
510 END
```

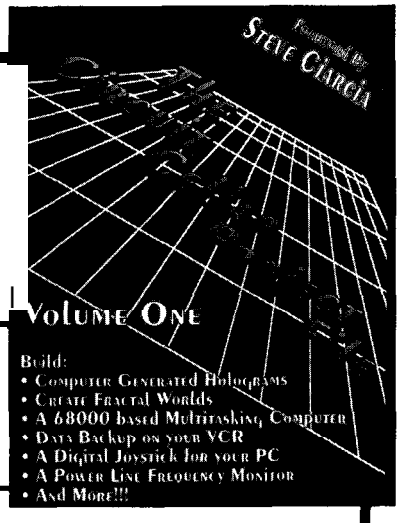
**listing 1**—*BASIC-52* makes a convenient front end for the *EconoRAM support* **routines written in assembly** language.

## TIME-SLOT SUPPORT ROUTINES

The bidirectional ports of the 8031 make it ideal for interfacing with EconoRAM. I chose the RTC52 board and the 80C52 as a platform because I could quickly write some BASIC code as a shell for the assembly language time slot routines. The standard 11.0592-MHz crystal has an execution cycle slightly longer than 1 µs. Although I could have used a timer for accurate timing loops, I chose to use software loops because absolute timing wasn't necessary.

Figure 2 shows the interface used with the DS2223 EconoRAM. It gives 32 bytes (256 bits) of storage, which is


The Circuit Cellar Project File, Volume 1 has over 200 pages of new or expanded hands-on projects and tutorials. Circuit Cellar INK's editors have chosen a dozen of the top projects from the Circuit Cellar Design Contest, independent submissions, and top-response articles to make a book with something for every interest. You'll get projects about:




- A VCR Data Backup Card
- A Digital Joystick Port for Your PC
- A Power-line Frequency Monitor
- Computer-Generated Holography
- An LCD Namebadge
- A Car-computer Diagnostic Tool
- An LCD Tester
- Fractals
- Front-Door Light Control
- An Electronic Combination Lock
- Building & Debugging 66000 Multitasking Computers

topped off with a foreword by Steve Ciarcia!
All Of this is yours for **only** $24.95 (includes domesticdelivery')
Order your copy today!

**NOW SHIPPING!**

**The Circuit Cellar Project File, Volume 1**
**4 Park Street • Vernon, CT 06066**
**Tel: (203) 8752199 • Fax: (203) 872-2204**

*$24.95 Visa, Mastercard, Check, or Money Order
(Add $2.00 for delivery to Canada or Mexico, $4.00 for delivery to other non-U.S. addresses)

```
        ORG    6000H
JSYNC:  LJMP   SYNC
JREAD:  LJMP   READ
JWRITE: LJMP   WRITE
JPRES:  LJMP   PRES
;
SYNC:   PUSH   B
        PUSH   DPL
        PUSH   DPH
        MOV    DPTR,#0208H ;COUNT TO 264
LOOP1:  ACALL  WRZERO      ;ZERO TIMESLOT
        DJNZ   DPL,LOOP1
        DJNZ   DPH,LOOP1
        POP    DPH
        POP    DPL
        POP    B
        RET
;
;
READ:   PUSH   ACC
        PUSH   B
        PUSH   DPL
        PUSH   DPH
        PUSH   PSW
        PUSH   00H
        ANL    PSW,#0E7H   ;WORKING BANK 0
        MOV    R0,#4DH     ;STACK SPACE
        MOV    DPL,#08H    :GONNA SEND 8 BITS
        MOV    A,#01H      ;READ COMMAND
LOOP2:  RRC    A           ;LSB INTO CARRY
        JNC    BRANCH1     ;IF A '0'
        ACALL  WROME       ;ONE TIMESLOT
        AJMP   BRANCH2
BRANCH1: ACALL WRZERO      ;ZERO TIMESLOT
BRANCH2: DJNZ  DPL,LOOP2   ;DO ALL BITS
        MOV    DPH,#20H    ;GET 32 BYTES
LOOP3:  MOV    DPL,#08H    ;GET 8 BITS
LOOP4:  ACALL  RDSOME      ;READ A TIMESLOT
        RRC    A           ;CARRY INTO MSB
        DJNZ   DPL,LOOP4   ;DO ALL BITS
        MOV    @R0,A       ;SAVE IN USER REG
        INC    R0          ;NEXT USER REG
        DJNZ   DPH,LOOP3   ;DO ALL BYTES
EXITRD: POP    00H
        POP    PSW
        POP    DPH
        POP    DPL
        POP    B
        POP    ACC
        RET
;
;
WRITE:  PUSH   ACC
        PUSH   B
        PUSH   DPL
        PUSH   DPH
        PUSH   PSW
        PUSH   00H
        ANL    PSW,#0E7H   ;WORKING BANK 0
        MOV    R0,#4DH     ;STACK SPACE
        MOV    DPL,#08H    ;SEND 8 BITS
        MOV    A,#0F9H     ;WRITE COMMAND
LOOP5:  RRC    A           ;LS% INTO CARRY
        JNC    BRANCH3     ;IF A '0'
        ACALL  WRONE       ;ONE TIMESLOT
        AJMP   BRANCH4
BRANCH3: ACALL WRZERO      ;ZERO TIMESLOT
BRANCH4: DJNZ  DPL,LOOP5   ;DO ALL BITS
        MOV    DPH,#20H    ;WRITE 32 BYTES
LOOP6:  MOV    DPL,#08H    ;WRITE 8 BITS
        MOV    A,@R0       ;GET BYTE TO WRITE
LOOP7:  RRC    A           ;LSB INTO CARRY
        JNC    BRANCH5     ;GO DO A WRZERO
        ACALL  WRONE       ;ONE TIMESLOT
        AJMP   BRANCH6      GO ON
BRANCH5: ACALL WRZERO      ;ZERO TIMESLOT
BRANCH6: DJNZ  DPL,LOOP7    DO ALL BITS
        INC    R0          ;NEXT USER REG
        DJNZ   DPH,LOOP6    DO ALL BYTES
EXITWR: POP    00H
        POP    PSW
        POP    DPH
        POP    DPL
        POP    B
```

(continued)

listing **2**—*The main timeslot routines are written in 8031 assembler and are called from BASIC-52.*


NOW DOUBLED IN LIBRARY SIZE **AND WINDOWS 3.0**

# DC/CAD

CAD Showdown Results!!

**HIGH DENSITY EXPERTS!**

**Integrates Schematic, PCB Layout & Autorouting**

This top-rated **CAD** out-routed the competition in the **1990** CAD Show-down. DC/CAD displayed its power and flexibility when routing a double-sided board while competing routers used four to six layers. This non-copy protected package with surface mount support includes:

CALL TODAY DC/CAD $595 (Priced from $395)

- Multi-strategy 1-mil parts autoplacer
- "1-mil" autorouting with rip up 8 retry
- Thorough annotating design rule checker
- Full P-way GERBER and DXF support
- Optional autoground plane support with cross-hatching
- Optional simulation capability & protected mode for 386 users

**LEASE** PROGRAM a SITE LICENSE AVAILABLE

30 DAY MONEY BACK GUARANTEE



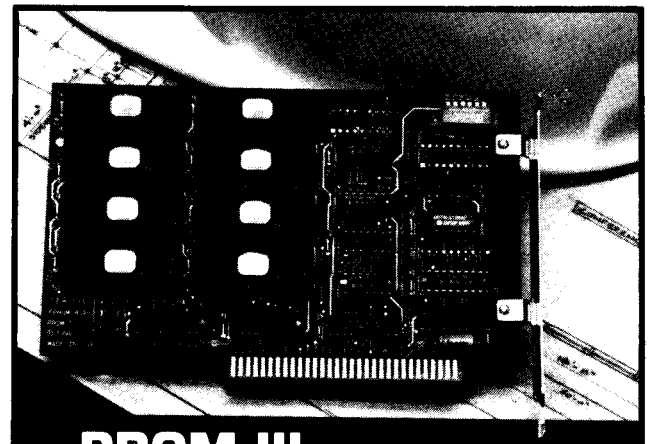
DESIGN COMPUTATION

Rt. 33, Sberman Square Farmingdale, NJ 07727

(908) 938-6661 • (908) 938-6662 (FAX)

*DC/CAD...Innovative, Intelligent and Integrated Software*

Reader Service #134



# PROM-III

- PUT DOS AND APPLICATION IN EPROM
- ALLOWS DISKLESS OPERATION
- UP TO 1 MBYTE ROM-DRIVE WITH 16K FOOTPRINT
- PROMKIT SOFTWARE BY ANNA-BOOKS
- FLASH EPROM SUPPORTED
- BATTERY RAM MODULES SUPPORTED
- DELIVERY FROM STOCK

SEALEVEL

COMMUNICATIONS & I/O

SEALEVEL SYSTEMS INC.
PO BOX 830
LIBERTY, SC 29657
(803) 843-4343

Reader Service #190

enough room for ten social security numbers (in BCD). A 30-mAH lithium battery will hold those numbers firm for over ten years. If the power source for the EconoRAM is always the battery, then the number of connections necessary falls from three to two. Figure 3 shows this implementation.

Now let's investigate the DS2223's sister part, the DS2224. Same interface, same communication protocol, same number of bits of storage, but the first 32 bits are ROM and not RAM. Not only are they ROM, but they are prewritten with a unique serial number. No two are the same, at least not for the first 4,294,967,295 parts made, and each one has its own signature ID. Even if you never use the RAM, this part is ideal for identification. Figure 4 demonstrates how the data line can be used to provide power for the DS2224. The capacitor retains enough of a charge to power communications without the usual third connection. Of course, it does not retain the data in the remaining 224 bits once the device has been disconnected. However, in this case, we are only interested in the 32 bits of ROM data, which will be safe even without power.

## DATA INTEGRITY

Data passing from device to host can be read incorrectly. A checksum of the 32-bit serial number and any other RAM data would provide greater data protection. A calculation and checksum comparison would reveal po-

```
          POP   ACC
          RET
;
,
PRES:     PUSH  ACC
          PUSH  B
          PUSH  DPL
          PUSH  DPH
          PUSH  PSW
          PUSH  00H
;
          ANL   PSW,#0E?H   ;WORKING BANK 0
          MOV   R0,#4DH     ;STACK SPACE
          ACALL PRESDET     ;LOOK FOR PRESENCE
          MOV   A,@R0       ;PRESENCE FLAG
          CJNE  A,#0FFH,EXITPR ;NO ONE HOME
          INC   R0          ;MOVE THE POINTER
          MOV   DPL,#08H    ;SEND 8 BITS
          MOV   A,#0FH      ;DS2400 COMMAND
LOOP8:    RRC   A           ;LSB INTO CARRY
          JNC   BRANCH7     ;IF A '0'
          ACALL WRONE       ;ONE TIMESLOT
          AJMP  BRANCH8
BRANCH7:  ACALL WRZERO      ;ZERO TIMESLOT
BRANCH8:  DJNZ  DPL,LOOP8   ;DO ALL BITS
;
          MOV   DPH,#08H    ;GET 8 BYTES
LOOP9:    MOV   DPL,#08H    ;GET 8 BITS
LOOP10:   ACALL RDSOME      ;READ A TIMESLOT
          RRC   A           ;CARRY INTO MSB
          DJNZ  DPL,LOOP10  ;DO ALL BITS
          MOV   @R0,A       ;SAVE IN USER REG
          INC   R0          ;NEXT USER REG
          DJNZ  DPH,LOOP9   ;DO ALL BYTES
,
EXITPR:   POP   00H
          POP   PSW
          POP   DPH
          POP   DPL
          POP   B
          POP   ACC
          RET
;
,
WRONE:    CLR   P1.l        ;OUTPUT A LOW
          SETB  P1.1        ;OUTPUT A HIGH
          MOV   B,#1CH      ;28 LOOPS (60 us)
          ACALL LOOP        ;DELAY
          RET
,
WRZERO:   CLR   P1.1        ;OUTPUT A LOW
          MOV   B,#1CH      ;28 LOOPS (60 us)
          ACALL LOOP        ;DELAY
          SETB  P1.l        ;OUTPUT A HIGH
          RET
,
RDSOME:   CLR   P1.1        ;OUTPUT A LOW
          SETB  P1.1        ;OUTPUT A HIGH
          MOV   B,#01H      ;1 LOOP (4 us)
          ACALL LOOP        ;DELAY
          MOV   C,P1.1      ;PUT BIT IN CARRY
          MOV   B,#1AH      ;26 LOOPS (56 us)
          ACALL LOOP        ;DELAY
          RET
,
PRESDET:  CLR   P1.l        ;OUTPUT A LOW
          MOV   B,#0ECH     ;236 LPS (480 us)
          ACALL LOOP        ;DELAY
          SETB  P1.1        ;OUTPUT A HIGH
          MOV   B,#05H      ;5 LOOPS (16 us)
          ACALL LOOP        ;DELAY
          MOV   A,#3CH      ;60 LOOPS (240 us)
          MOV   @R0,#00H    ;NO PRESENCE
LOOP11:   JB    P1.1,BRANCH11  ;J IF NO LOW
          MOV   @R0,#0FFH   ;FLAG PRESENCE
BRANCH11:DJNZ   ACC,LOOPll  ;KEEP LOOKING
          RET


LOOP:     DJNZ  B,LOOP            ;DELAY LOOP
          RET


END:      END
```

**Listing *P-continued***


smx simple multitasking executive V2.1

FULL FEATURED KERNEL

- task manager
- memory manager
- intertask comm.
- error manager
- i/o, events, & timing
- resource manager
- preemptive
- ROM'able

EASY TO USE

- libraries for Microsoft C, Borland C/C++, Zortech C/C++ and assembler
- Quick Start, User's Guide, and Reference manuals
- standalone, PC, and DOS platforms
- *smxProbe* task debugger
- 6 months free support and updates

FAST & SMALL

- 15 usec max. interrupt latency
- 150 usec typical task switch
- 8 to 25 KB code size

**Ask about our $95 evaluation kit.**

MICRO DIGITAL
6402 Tulagi St.
Cypress CA 90630-5630

**Call for information:**
**1-800-366-2491**
FAX 714-891-2363

Reader Service #165

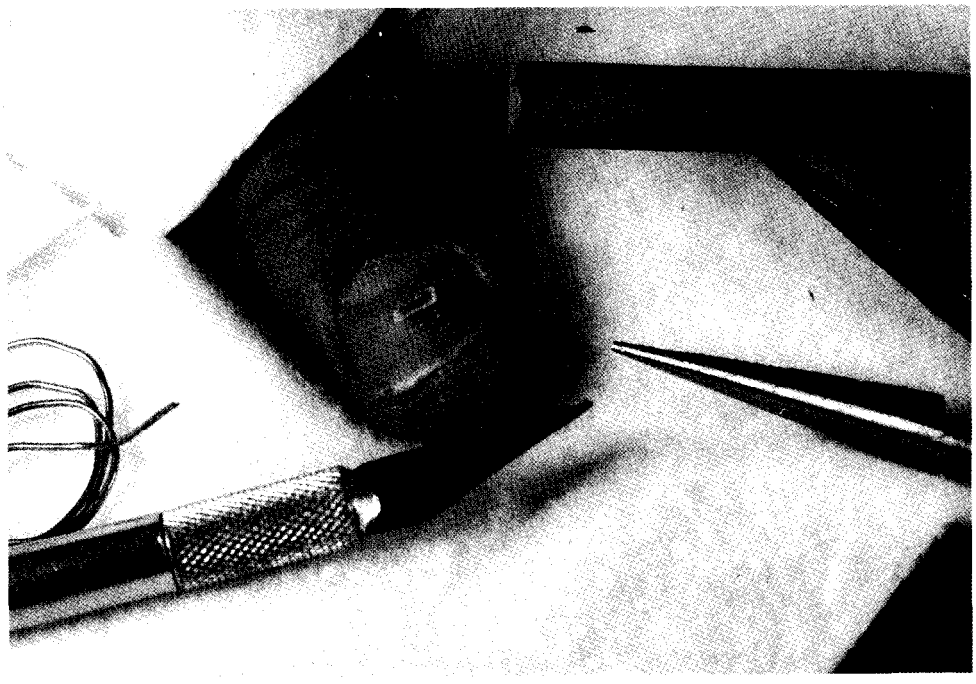


**Photo 1** *-Perhaps by the year 2005, we'// have a less painful way to Implant the thumbnail PIN than a knife and soldering iron.*

tentially garbled data upon the reading of the device. However, inorder for a calculated checksum to be stored along with any other data, the RAM bits must be battery backed.

Enter a new device. The DS2400 silicon serial number extends the possible bit combinations from 32 bits to 48 bits of prewritten ROM. An B-bit CRC is appended to the serial number ensuring data integrity. This device is strictly ROM and needs no battery backup. Time slots are used for data communication, as with the DS2223 and DS2224, but the DS2400 has a unique feature. If the host sends a "zero" time slot of 480-µs minimum duration, any connected DS2400 will answer with a presencesignal. If the host detects the presence of a DS2400 and sends the command byte OFH, the DS2400 will return the type identifier byte 01H, followed by the 48-bit serial number and B-bit CRC. The presence signal and type identifier set the groundwork for a whole family of devices that are sure to come off the drawing boards of Dallas Semiconductor.

Listing 1 is the BASIC shell I wrote for the RTC52. *This* program makes calls to the time-slot routines (Listing 2) written in machine language. The first routine writes 264


NEW VERSION AVAILABLE

*The Heart of the Matter...*

**RTXC™**

**Real-Time Multitasking Executive**

. INTEL 80x88/86, 8096/80C196
. MOTOROLA 680x0, 683xx, 68HC11,68HC16
. INMOS T400, T800 . ANALOG DEVICES 2100

- Preemptive Scheduling
- Fixed or Dynamic Priorities
- Time out on some services
- Configurable and ROMable
- Intertask Communcations
  - Messages
  - Queues
  - Semaphores
- Memory Management
- Resource Manager
- Over 40 Executive Services
- System Level Debugging Utility
- System Generation Utility
- Written in C
- Source Code Included
- No Royalties
- Technical Support
- Widely Ported
- Sensible License Agreement
- Most Popular C Compilers Supported

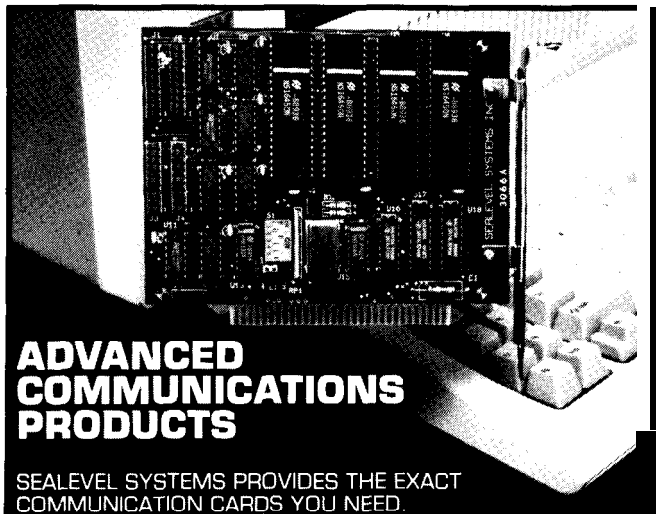
FREE DEMO AVAILABLE

One Time License Fee $2,995
Discounts for Multiple Licenses/Ports
*The only real-time kernel you'll ever need™*

PC SUPPORT AVAILABLE

**A.T. BARRETT & ASSOCIATES**
**11501 Chimney Rock, Houston, TX 77035**
**FAX 713/728-1049**
**Phone 800/525-4302**

Reader Service #109



**ADVANCED COMMUNICATIONS PRODUCTS**

SEALEVEL SYSTEMS PROVIDES THE EXACT COMMUNICATION CARDS YOU NEED.

PRODUCTS:

- 1, 2 OR 4 PORT RS-232 AND RS-422/485 BOARDS
- CURRENT LOOP SERIAL INTERFACES
- HIGH SPEED SYNC (HDLC, SDLC) AND ASYNC WITH DMA
- RS-530 AND V.35 INTERFACE BOARDS
- DIGITAL AND RELAY I/O BOARDS.
- DISKLESS EPROM BOARD WITH PROMKIT SOFTWARE BY ANNABOOKS.
- NEW LAP-TOP ADD ONS! • MADE IN USA
- DELIVERY FROM STOCK • SATISFACTION GUARANTEED
- EXCELLENT TECHNICAL SUPPORT

SEALEVEL COMMUNICATIONS & I/O

SEALEVEL SYSTEMS INC.
PO BOX 830
LIBERTY, SC 29657
(803) 843-4343

Reader Service #191

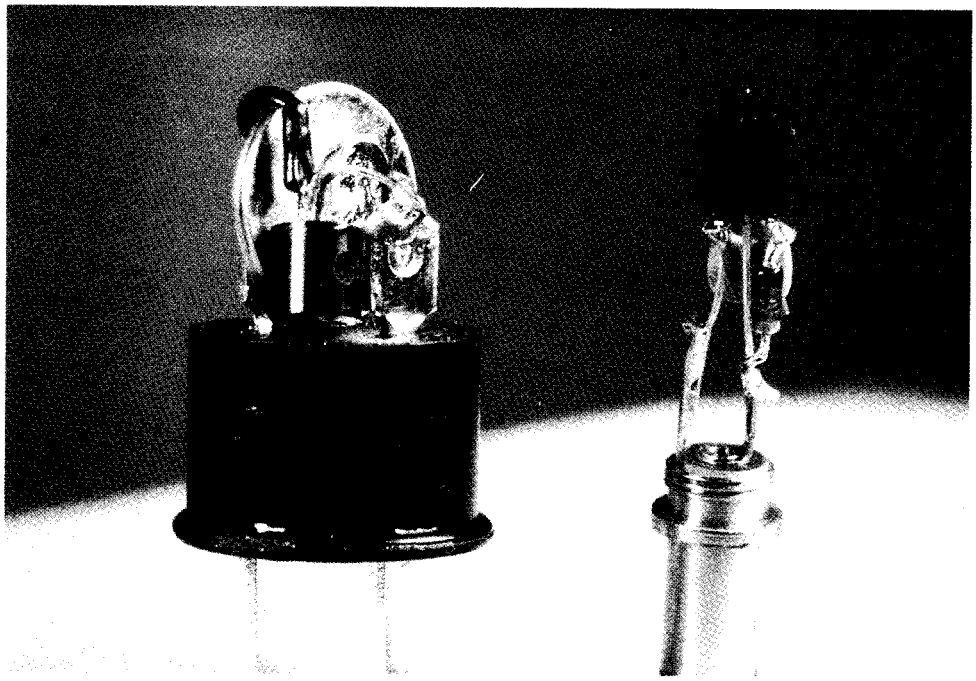


**Photo 2**—*The EconoRAM's small size* lets you ***install it inside a connector housing. On the left, an EconoRAM is installed on an Amphenol connector with its own button cell battery. On the right, a data-line-powered EconoRAM fits on a typical power connector.***

"zero" time slots to sync up the DS2223 or DS2224. The second reads 256 bits from a DS2223 or a DS2224. The third writes 256 bits to a DS2223 or a DS2224. (The data written to the ROM bits are tossed into the bit bucket.) Finally, the fourthroutinelooks for a DS2400 and reads the 48-bit serial number and B-bit CRC. BASIC then calculates a CRC and compares it with what was read, verifying a good data transmission.

I'm sure you can see all that these units make possible. Low cost and interface simplicity allow easy integration of these devices into many applications. Some uses could involve home or auto security, inventory and product identification, bank transactions with running account information, and even an implanted PIN. Just don't expect to get a duplicate made at the local hardware store.

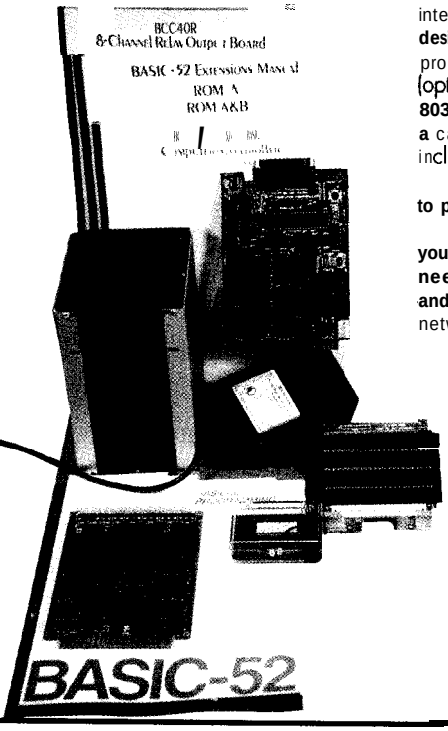
*Jeff Bachiochi (pronounced "BAH-key-AH-key") is an electrical engineer on the Circuit Cellar INK engineering staff. His background includes product design and* ***manufacturing.***

**IRS**

419 Very Useful
420 Moderately Useful
421 Not Useful


DON'T TELL THE BOSS

that using Micromint's new C-PAK Controller Development Package made solving that closed loop control problem a cinch. Let him think you sweated over it for weeks!




Micromint now offers engineers and programmers a convenient, low-cost combination development and application hardware system intended to act both as the development tool and the hardware solution for closed-loop control designs. In one package we give you a high-speed 80C52 microcontroller with built-in EPROM programmer, console serial port, serial printer port and 24 bits of bidirectional parallel TTL (optoisolated input optional); both a floating-point BASIC interpreter (compiler optional) and an 8031 assembler; a BASIC extensions and utilities ROM; eight 2-amp relays for control outputs; a card cage; 4-slot motherboard; and a power supply. We also throw in lots of manuals, including the latest BASIC-52 Proarammer's manual.*

According to our current cot&g, all these pieces odd up to $591, but Micromint wants to put a C-PAK in your hands for the unheard of price of $379!

Go through our catalog to check the exact specifications or fax us a request and we'll send you the data sheets for the C-PAK. C-PAK is also expandable, so consider some options if you need more I/O. We offer bus-compatible boards containing more parallel I/O, optoisolators and drivers, relays, 8-bit and 12-bit ADCs, more serial ports, displays, disk drives, and even a network. Start small but think big. You'll find that there isn't a better controller value on the market.

Micromint C-PAK Special $379 plus shipping. To include the HOST-52 software development package for the IBM PC add $50.

*C-PAK contents. BCC52 w/8K RAM, ROM A&B assembler/utilities, BCC40R, CC03 cord cage, MB04 motherboard, BASICMAN, PS11, and manuals. Micromint reserves the right to limit quantities to any single customer and to terminate this offer at any time.

MICROMINT, INC.
4 PARK STREET
VERNON, CT 06066
Tel: (203) 871-6170
Fax: (203) 872-2204

Reader Service #166

# Summer Harvest

*A Cornucopia Of Chips*

**SILICON UPDATE**
Tom Cantrell

There was a time when summer in the not-yet-Silicon Valley was a rather leisurely affair. Not so these days when "Summer Slowdown" refers to the cyclical dip in IC sales, not sitting on the porch with an iced tea watching the fish jump and the cotton get high.

IC sales may slow down while the nation's purchasing managers drag their kids to Wally World, but the development of new and wondrous ICs doesn't. Here's a sampling of Silicon Valley's summer harvest,

**HACKER PARADISE**

Where's a die-hard computer guru supposed to go on vacation anyway? Remember, these are people who take their kids to computer camp, not Wally World. In fact, given the propensity of the most obsessed to avoid daylight at all cost, summer is just a good excuse to spend even more hours in the lab. However, the preconception that computer types are antisocial is overblown-they just prefer to hang with their own kind. Thus, a "fun get-away" like the IEEE Hot Chips symposium has no trouble attracting hundreds of chip-heads, including me, from around the world.

Check out the fun-filled schedule for a day at Hot Chips. All I can say is, watch out Club Med!

| | |
|---|---|
| 8:00–10:00 | Listen to people talk about chips |
| 10:00–10:30 | Break-talk about chips over coffee |
| 10:30–12:00 | Listen to people talk about chips |
| 12:00–1:00 | Lunch-talk about chips over food |
| 1:00–3:00 | Listen to people talk about chips |
| 2:30–3:00 | Break-talk about chips over colas |

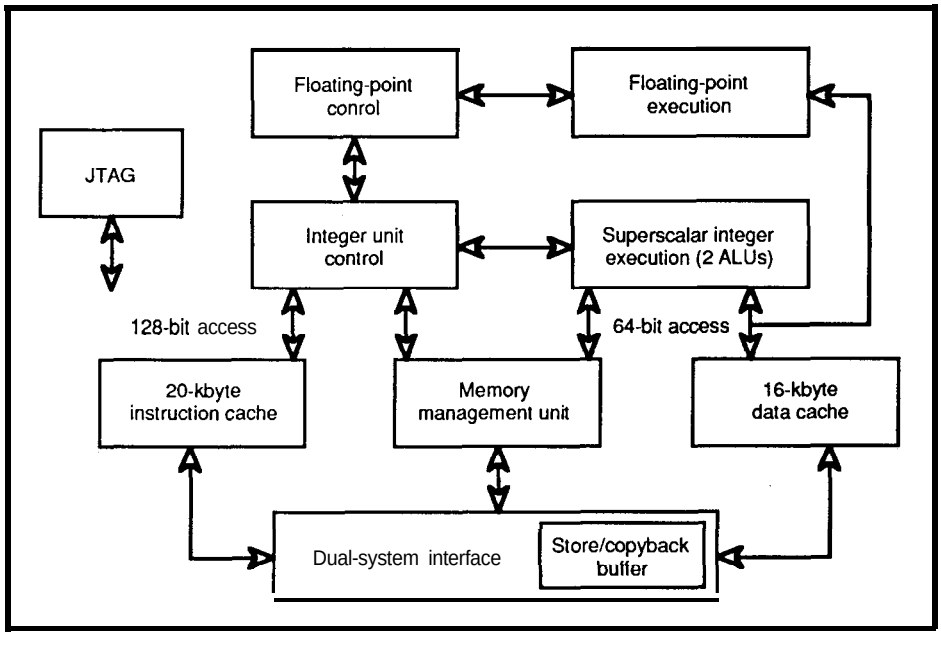



Figure 1—*The upcoming SuperSparc superscalar chip from Sun and TI will have two integer units.*

**Complexity Considerations in Big-Issue Machines**

- Operand bypassing grows as

$$(\text{no_of_src_operands})\,(\text{results_in_pipeline})\,i = 2\ (is)\ i = 0\ \left(i^2\right)$$

- Data dependence checks grow as

$$(\text{no_of_src_operands}) \sum_{n=1}^{i-1} n = 2\left(\frac{i(i-1)}{2}\right) = O\left(i^2\right)$$

- **Plus** priority encoder complexity
  – additionally increases logic complexity or logic depth

- Complexity grows fasterthan issue rate

**Figure 2**—*The feasibility of speculative execution relies heavily on the conventional wisdom that 'transistors are free." However, many note that 'wires aren 't."*

| | |
|---|---|
| 3:30–5:00 | Listen to people talk about chips |
| 5:00–7:00 | Reception-talk about chips over wine |
| 7:00–10:00 | Listen to people talk loudly (good `wine!`) about chips |

Despite all the talk, there wasn't much new. In fact, as I claimed in an earlier article that logic synthesis represents the "End Of Hardware," it seems we've just about reached the "End Of Architecture."

These days the main controversies concern two questions: What is the best way to exploit instruction-level parallelism, and is there any instruction-level parallelism to exploit?

The interesting point is the questions are being addressed in the above order; that is, people have been beating their brains out over the first question for some time, but only now are they taking a close look at the second question.

Instruction-level parallelism refers to the fact (hope?) that "sequential" instructions can be packed together for simultaneous execution by a suitably equipped machine. For instance, there isn't any reason

```
A(i)=B(i)      ; execute one
A(i+100)=C(i) ;  after the other
```

can't be "parallelized" into

```
A(i)=B(i) :  A(i+100)=C(i)  ; execute both
                            ; at the same
                            ; time
```

But how about

```
A(i)=B(i)
A(i+j)=C(i)
```

On first glance it looks like they could be combined similarly, but don't forget the case where $j$=0 which will cause the machine to try to assign two different values to $A(i)$ at once!

Researchers seem to disagree on the answer to the second question (i.e., how much parallelism is present in real-world programs?). Various studies report results ranging from 1.37 to 90! Needless to say, research continues...

Anyway, since it's greater than one, we'll be seeing more superscalar chips with two integer units like the presumably soon-to-appear SuperSparc (previously known as "Viking") from Sun and TI (Figure 1). Supposedly Intel had a similar idea in mind for the '586, but has scrapped it (R&D budget shifting to the legal department?). I assume it may resurface in a later 'x86.

The problem is, superscalar-which attempts to parallelize your program while it is running-leads to horrible circuit complexity. Someone put a transparency up during the "Five Instructions Per Clock: Truth Or Consequences" session (after the wine!) that said "Superscalar-The CISC Empire Strikes Back."

So what's next? Well, while speculation may have reigned supreme on Wall Street in the '80s, it's moving to Silicon Valley in the '90s: "speculative execution" is the latest hot-button. For instance, if you have two execution units and encounter a conditional branch, don't wait for the condition to be resolved, but simply continue execution down both paths. After the condition becomes clear, side effects from the incorrect path are removed. The feasibility of speculative execution relies heavily on the conventional wisdom that "transistors are free." But it has been noted by some, including Michael Smith of Stanford (Figure 2), that "wires aren't."

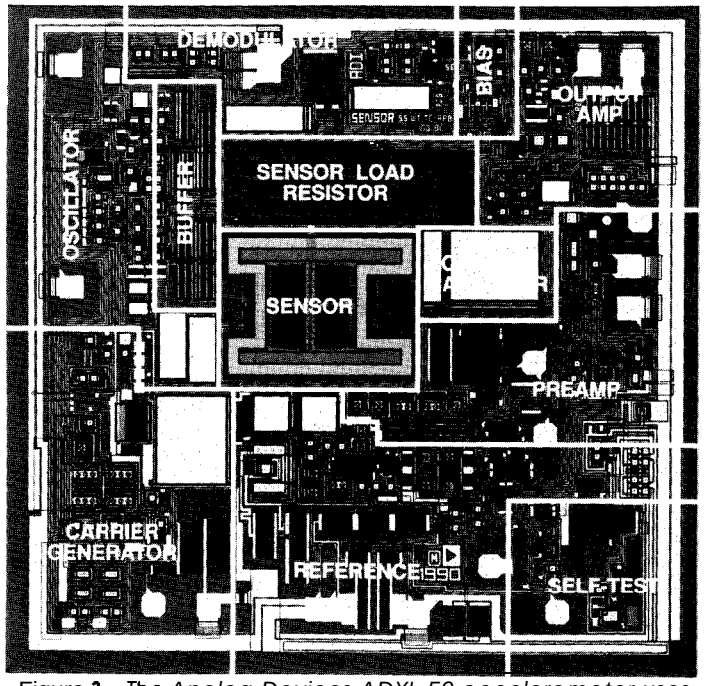



Figure 3—*The Analog Devices ADXL-50 accelerometer uses micromachining techniques to measure acceleration in a single plane from 0 to 50 g's.*

Kind of makes you long for the days of BASIC on a 6502, huh? That's how I know we've really reached the "End Of Architecture."

Overheard: Two gurus from secret startup XYZ Corp. comparing their new whizzy to market leader ABC's latest widget:

XYZ Guru #1—"We're way ahead of ABC"

XYZ Guru #2—"Yeah, all we've got to do is get it working..."

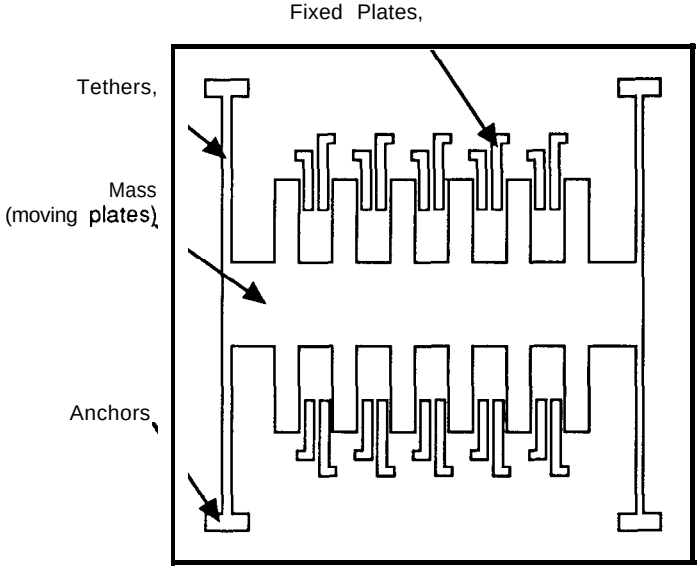



**Figure 4**—*The ADXL-50's "H"-shaped micromachined sensor located in the middle of the die is the mass that is acted upon by acceleration, while interleaved capacitors detect the change.*

## MICROMACHINES

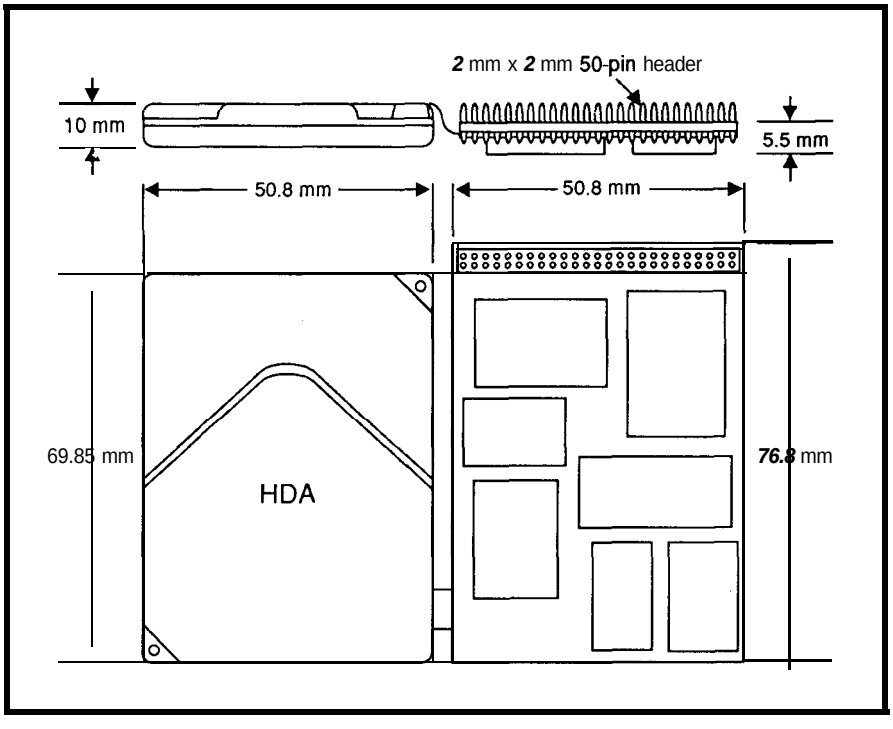
In my opinion, the real secret for high performance is "crank the clock," but such a view would certainly be deemed technically incorrect by the supertypes. Nevertheless, it seems the most interesting developments (like 100-MHz CMOS) are in the hands of semiconductor process wizards.

Beyond high speed, the wizards have even more tricks up their sleeves: micromachines. Micromachines result when semiconductor processes are used to make electromechanical devices. For instance, researchers have already demonstrated prototypes of tiny chip motors and gears.

On a more practical front, micromachining techniques are already being put to work by Analog Devices in their new ADXL-50 accelerometer (Figure 3). The device measures acceleration in a single plane from 0 to 50 g's with 5% accuracy and is designed especially for automotive airbag systems that currently use multiple mechanical sensors. However, the ADXL50 can certainly be used for other applications, and they will benefit from the competitive pricing-$5.00 in high volume-dictated by the automotive marketplace.

Figure 4 shows a close-up of the "H''-shaped micromachined sensor located in the middle of the die. The "H''-shaped portion is the mass that is acted upon by acceleration, while interleaved capacitors detect the change. Actually, the device is operated in a Force/Balance mode in which an opposing electrical force is applied to keep the mass stationary. The amount of this charge corresponds to the acceleration, and additional on-chip circuits further condition and amplify the output.

Figure **5**—*Most hard disk suppliers ore currently working on 2.5" models, but Integral* Peripjerols ***Inc.*** is leapfrogging ***everyone else and showing a 1.8" drive.*** *Initial* ***offerings include*** *20-megabyte units,* ***buf plans for*** *40-megabyte* ***units ore under way.***



The ability to combine electronic and mechanical functions on the same die is further exploited to incorporate a self-test feature that allows the airbag controller to test the sensor by initiating electrostatic deflection of the mass and checking for the appropriate output voltage. Needless to say, this is much better than finding out the hard way whether your 5-10-year-old airbag still works.

Micromachines promise some interesting applications in the future. I personally look forward to the proposed "Roto Rooter" micromachines that, after being injected into the bloodstream, will trundle along chomping up the artery-clogging byproducts of the Silicon Valley lifestyle. Oh, hang on a second, the pizza delivery guy is here...

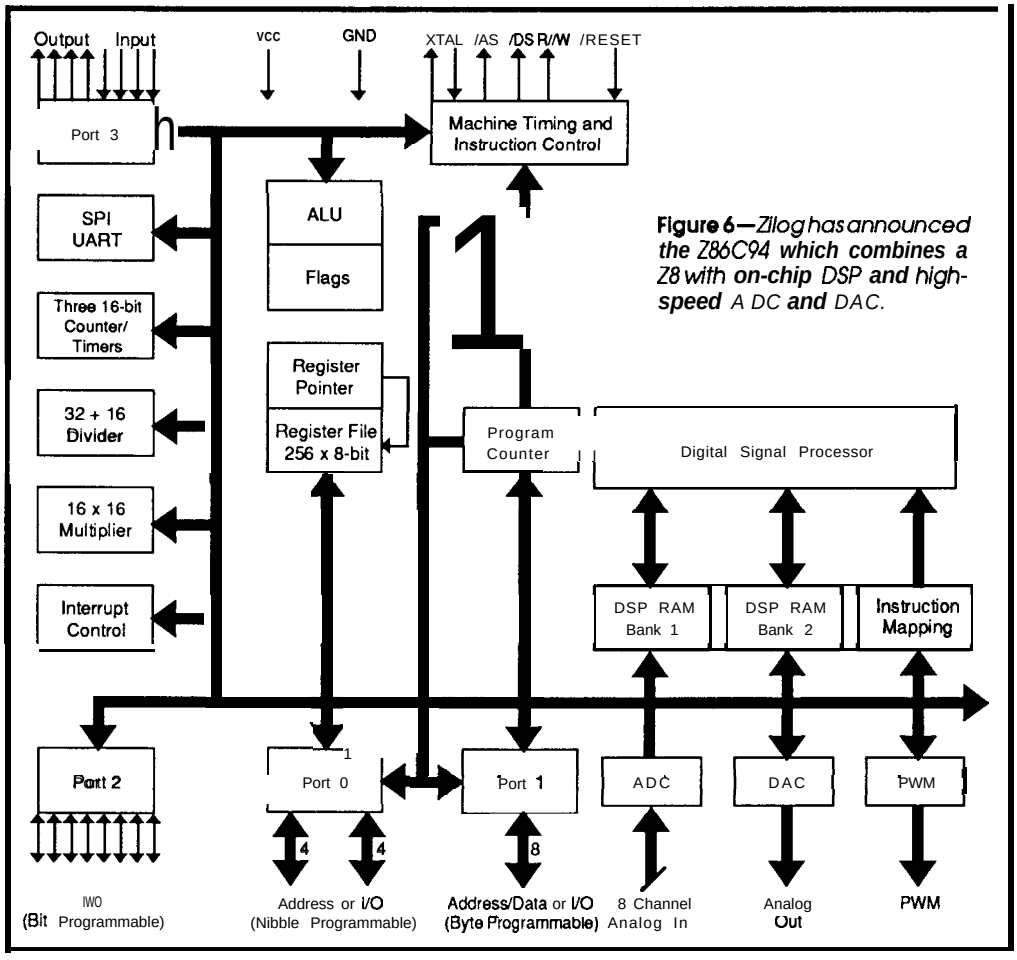
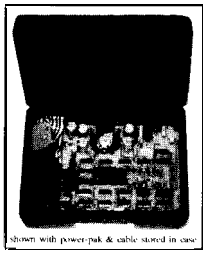



**Figure 6**—*Zilog has announced the Z86C94 which combines a Z8 with on-chip DSP and high-speed A DC and DAC.*


# MS-DOS EPROM PROGRAMMING SYSTEM NEEDS NO INTERNAL CARD

EPROMS (24, 28, 32 and 40* pins)
2708, 2758, TMS2716*, 2716
27C16, 2516, 2532*, 2564*
68764*, 68766*, 2732, 2732A
27C32, 2764, 2764A, 27C64
27128, 27128A, 27C 128
27256, 27C256, 27512
27C512, 27513*
27011*, 27C011* (1 MEG)
27010, 27C010
27C1000, 27C1001
27C020, 27C2001 (2 MEG)
27C040, 27C4001 (4 MEG)




EEPROMS
2804, 2816A, 28C16
2817A*, 2864A, 28C64
28256, 28C256, 52B13*
52B33*

*MicroControllers*
8741A*, 8742*, 8748(H)*
8749(H)* 8751*, 87C51*
8752* 8753*, 8744*
68705 *, TMS7742
*ADAPTER REQUIRED
Diagrams included with manual
Assembled adapters are available

*CONNECTS TO YOUR SYSTEM'S*
**PARALLEL PRINTER PORT**

A FAST, EASY-TO-USE SYSTEM WORKS WITH ANY DESKTOP OR LAPTOP MACHINE
EXPANDABLE, FLEXABLE DESIGN SUPPORTS CURRENT AND FUTURE DEVICES
SUPPORTS 8, 16 & 32 BIT DATA FORMATS INCLUDING WORD SPLIT & DOUBLE WORD
READS AND CONVERTS INTEL HEX. MOTOROLA S-RECORD AND BINARY FILES
HARDWARE PROTECTED AGAINST DEFECTIVE AND INCORRECTLY INSERTED DEVICES
NO SOFTWARE INSTALLATION NECESSARY. PROGRAM TOTALLY SELF CONFIGURING

## SYSTEM SOFTWARE COMMANDS

- PROGRAM EPROM(S) FROM DISK FILE
- READ DISK FILE INTO BUFFER
- READ EPROM(S) INTO BUFFER
- SAVE EPROM(S) OR BUFFER TO DISK
- PROGRAM EPROM(S) FROM BUFFER
- COMPARE EPROM(S) WITH BUFFER
- COPY EPROM(S)
- VERIFY EPROM ERASED
- SELECT BUFFER EDITOR
- SELECT DEVICE TYPE
- DEVICE CHECKSUM
- SET BUFFER (0, 1, 2, 3)

*PLUS AN INTEGRATED BUFFER EDITOR WITH 18 BYTE LEVEL COMMANDS*

SYSTEM INCLUDES: PROGRAMMING UNIT, POWER PACK CONNECTING CABLE. OPERATION MANUAL & SOFTWARE $289

SOFTWARE AVAILABLE ON 3 1/2" OR 5 1/4" DISK (PLEASE SPECIFY)
CALL ABOUT OPTIONAL ADAPTERS — A SOFT TRAVEL CASE IS AVAILABLE FOR $19.00

TO ORDER SEND CHECK, MONEY ORDER, WRITE OR CALL:

VISA

ANDRATECH
P.O. BOX 222
MILFORD, OHIO 451.50
(513) 831-9708
FAX (513) 831-7562

MASTER CARD

ADD $5.00 FOR SHIPPING

ADD $4.00 FOR C.O.D

WRITE FOR MORE INFORMATION OR CALL AND LISTEN TO OUR "TALKING" DATA SHEET

Reader Service #127

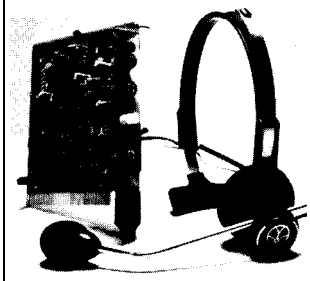


# TALK TO YOUR COMPUTER

WITH VOICE MASTER KEY@ FOR PCs/COMPATIBLES

***VOICE RECOGNITION WITH SPEECH RESPONSE***

processing,
Instant



Digitally record your own speech, sound. or music to put into your own software programs. Software provides sampling rate variations, graphics-based editing, and data compression utilities. Create software sound files, voice memos, more. Send voice mail through LANs or modem. A superior speech/sound development tool.

INTERACTIVE SPEECH INPUT/OUTPUT Tag your own digitized speech files to voice recognition macros. Provides speech response to your spoken commands -- all from within your application software! ideal for business, presentation, education, or entertainment programs you currently use. Augment the system for wireless uses in robotics, factory process controls, home automation, new products, etc. Voice Master Key System does it all!

EVERYTHING INCLUDED Voice Master Key System consists of a plug-in card, durable lightweight microphone headset, software, and manual. Card fits any available slot. External ports consist of mic inputs and volume controlled output sockets. High quality throughout, easy and fun to use.

ONLY $149.95 COMPLETE

ORDER HOTLINE: (503) 342-1271 Monday-Friday 8 AM to 5 PM Pacific Time. VISA/MasterCard phone or FAX orders accepted. No CODs. Personal checks subject to 3 week shipping delay. Specify computer type and disk format (3 1/2" or 5 1/4") when ordering. Add $5 shipping charge for delivery in USA and Canada. Foreign inquiries contact Covox for C & F quotes.

*30 DAY MONEY BACK GUARANTEE IF NOT COMPLETELY SATISFIED.*

CALL OR WRITE FOR FREE PRODUCT CATALOG.

**COVOX INC.**
675 CONGER ST.
EUGENE, OR 97402

TEL: (503) 342-1271
FAX: (503) 342-1283

Reader Service #129

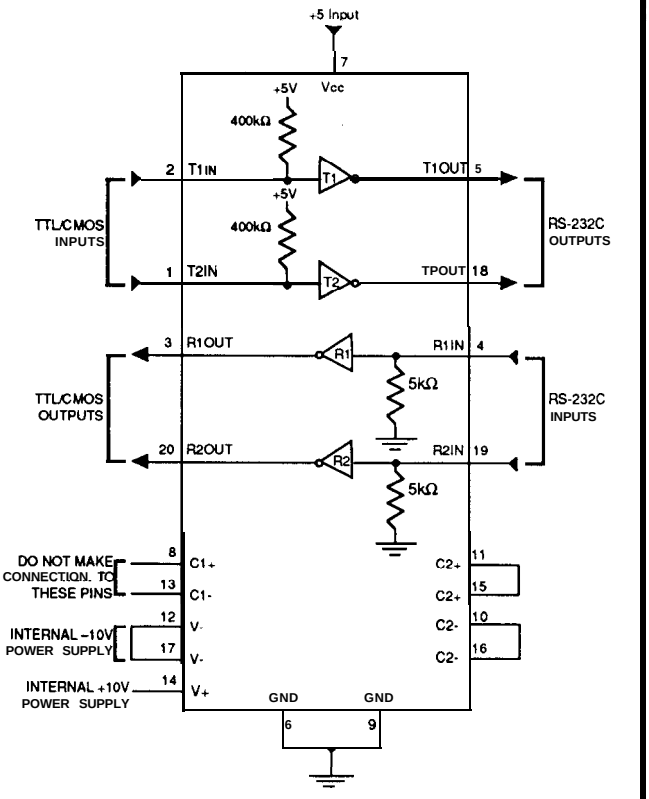



Figure 7—*The MAX233 includes the capacitors within the chip to minimize parts count.*

## MICRODISKS

One of the more interesting developments expected to surface at Comdex are tiny hard disks. Most suppliers are working on 2.5" models, but already one start-up—Integral Peripherals Inc.-is leapfrogging everyone else and showing a 1.8" drive (Figure 5). Initially they're offering a 20-megabyte unit, but plan to introduce 40-megabyte units soon. Incredible, isn't it?

These tiny drives are a natural for the emerging RAID-"Redundant Array of Inexpensive Disks"—concept. Interestingly, one of the pioneers of RAID research is David Patterson of Berkeley RISC fame. I consider the fact a leading CPU architect is switching to disk research more backup for my "End Of Architecture" hypothesis.

Anyway, the RAID concept calls for the use of multiple disk drives to increase performance (i.e., parallel access) and/or increase reliability (via redundancy). Different RAID configurations offer various tradeoffs between cost, performance, and fault tolerance.

Using the new 1.8" disks, I can imagine a "RAIDcard" that would combine multiple disks, controller electronics, and some RAM cache on a single plug-in board.

Beyond that, who knows? Maybe someone will make an even tinier disk that looks/plugs-in just like a RAM chip. Now that's virtual memory.

Note how, for packaging flexibility, the controller card and actual hard disk assembly (HDA) are physically


## BTK52 BASIC-52 TOOLKIT

The BTK52 is an intelligent front end for program development on the MCS BASIC-52 CPU It reduces 8052 program development time substantially and can be used with any MCS BASIC-52 based target system. The BTK52 runs on any IBM-PC/XT or compatible.

- *Program download from PC host to target*
- *Program upload from target to PC host*
- *BASIC program renumber utility, with "from," "through," "start," and "increment"*
- *Full screen program editing*
- *Single line editing with automatic error line number detection*
- *Full on-line help facility*
- *Transparent, adaptive line compression for full input line buffer utilization*
- *All functions accessible wih only one keystroke from the terminal emulator*
- ***$125***

## BXC51 805118052 BASIC COMPILER

- **Fully** ***compatible*** *with code written for MCS BASIC-52 interpreter*
- *Now with integer, byte and bit extensions for code that runs more than 50 times faster than the MSC BASIC-52 interpreter*
- *Full floating point support*
- *In-line assembly language option*
- *Compile time switch to select 805 1/8031 or 8052/8032 CPUs*
- *Includes Binary Technology's SXA-5 1 cross-assembler and Hex file manipulation utility*
- *Compatible with any RAM or ROM memory mapping*
- *Runs on IBM-PC/XT or compatible*
- ***$295***

**603-469-3232 • FAX 603-469-3530**

Binary Technology, Inc.

Main Street • P 0 Box 67 . Meriden NH 03770

Reader Service XI 13

## IMAGE PROCESSING

### *Victor Library for C programmers*

Image Processing: bright/contrast, sharpen, outline, resize, overlay, matrix conv, etc. TIFF/PCX/GIF/bin, use exten'd, expan'd, conv mem, images up to 4048 x 32768 grayscale, color, EGA/VGA up to 1024x768x256, Laserjet, ScanJet+, for MSC, QuickC, Turbo C/c++, BC++. Includes free copy of ZIP & extensive examples. Source avail. No royal. $195.

### *ZIP Image Processing software*

Bright/contrast, sharpen, outline, noise removal, embossing, matrix conv, etc. TIFF/PCX/GIF/EPS/bin. Up to 4048~4048, outstanding display and printing of grayscale images. EGA/VGA/super VGA, LaserJet, dot matrix. Versions for ImageWise/Idec/HRT/HP ScanJet. Source avail.

### *ZIP Colorkit*

Color reduction software, converts 24-bit TIFF and 16-, 24-, and 32-bit Targa images to EGA/VGA/super VGA. Lets any grayscale digitizer create color images. TIFF/PCX/GIF/TGA. $99.

### *Frame grabber*

Capture 512x512x256 NTSC or PAL "live" video on VGA, frame averaging. With Victor and ZIP Image Processing. $499.

*Catenary Systems*

*470 Belleview St Louis MO 63119*

**Call (314) 962-7833 to order**

VISA/MC/COD

Reader Service #118

separate. Even with a separate controller, tiny disks call for ever more highly integrated control ICs. Integral use a version of National Semiconductor's HPC family with special enhancements (multiply/accumulator) for disk motor (spindle and head) control.

Along similar lines, Zilog has announced the Z86C94 (Figure 6) which combines a Z8 with on-chip DSP and high-speed ADC and DAC. Like the National chip, the added circuits are designed for high-speed digital servo-motor control.

KISS

"Keep It Simple, Stupid" ranks right up there as one of my guiding precepts (along with "Don't Fix It If It Ain't Broke" and "S##t Happens"). Thus, I'm just as excited about plain-jane real-world problem solvers as I am about the latest technowidget.

I discussed one of these in a previous article: the Maxim MAX232 RS-232 driver/receiver chip. Its simple claim to fame was +5V-only operation. Before the MAX232, a triple-output power supply (+5V, ±12V) or kludgy DC-DC converter setup was required.

The **only thing** "broke" with the MAX232 is that it needs five external capacitors. Well, the new MAX233 (Figure 7) is finally the dream RS-232 chip I've been waiting for. Especially handy for tinkerers, where every extra connection potentially leads to late-night hair-pulling sessions (have you ever tried to debug a capacitor?).

**CONTACT**

Analog Devices Literature Center
70 Shawmut Rd.
Canton, MA 02021
(617) 937-1428

Zilog
210 E. Hacienda Ave.
Campbell, CA 950086600
(408) 370-8000

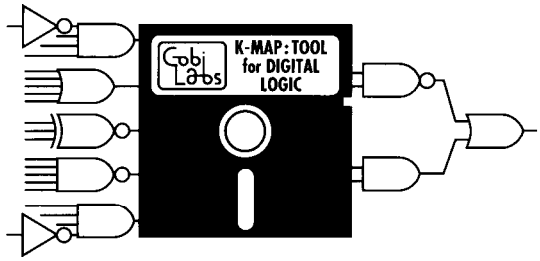
Maxim Integrated Products
120 San Gabriel Dr.
Sunnyvale, CA 94086
(408) 737-7600

*TomCantrell holds a B.S. and an M.B.A. from UCLA. He owns and operates Microfuture, Inc., and has been in Silicon Valley for ten years working on chip, board, and system design and marketing.*

**IRS**

422 Very Useful
423 Moderately Useful
424 Not Useful

DIGITAL DESIGNERS
MINIMIZE YOUR LOGIC




KARNAUGH MAP SOLVER FOR PC's
*Minimum Gates from Your Specs*
. **Combinational, Sequential**
. **Mealy** & Moore State Machines
. 2 to14 Inputs, **Any Number Outputs**
. **Sum of Products, Product of Sums**
. **JK, D, SR, and T Flip Flops**
. **Expert Reference and Tutorial**
• **30 Day Money Back Guarantee**
SAVE 40% : **$110** **+$5 (P&H)**
***FREE Demo Disk!***
**GOBI LABS: (407) 297-0862**
**BOX 616304, ORLANDO, FL 32861-6304**
Reader Service #144

What 8051 compiler would Santa use to automate his workshop?



BCI51™ of course - even Santa knows BASIC!
BCI51™ BASIC cross compiler with integer math • Supplement BASIC-5; or stand-alone • Now includes 'C51F support . BC151 with assemble1 and utilities $299 . Powerful Dallas DS5000T extensions $149 Assembly Language Programmer's Toolkit $149 ● superbdocumentatior
Systronix Inc.
754 East Roosevelt Avenue
Salt Lake City, Utah 84105
301-487-7412 FAX: 801-487-3130
Reader Service #197

# Affordable 8031 Development

## Single Board Computers, Assemblers, Compilers, Simulators, and EPROM Emulators

### Control-R Series, Single Board Computers

Two models of Control-R series computers make prototyping, one of a kind products, or small production runs easy and economical. Both feature RS232 compatible serial ports, single 5 volt supply operation, and direct access to Ports 1 and 3 of the 8031. Additional features are as follows:

### Control-R Model 1 $49.95

Fully populated board with I/O header for Ports 1 and 3, serial port, and 8K EPROM socket. 3.0" x 4.0"

### Control-R Model 2 $79.95

Same features as the Contml-R 1 plus 8K of SRAM and expansion bus with data, address, RST, INT1, WR, RD, PSEN, ALE and T1. 3.5" x 4.5"

### Software and Hard- Development Tools

### Control-C 8031 Cross-Compiler $200.00

The Control-C 8031 cross-compiler is a full featured K&R style C development system available at an affordable price. Optimized for embedded system use, it will produce ROMable code for any 8051 based system including designs using only the 128 bytes of internal RAM. Package includes compiler, pre-processor, assembler, simulator, printed documentation and complete library source code. Requires IBM PC or compatible. 5.25", 360K disk.

### PROMulator 256 $189.95

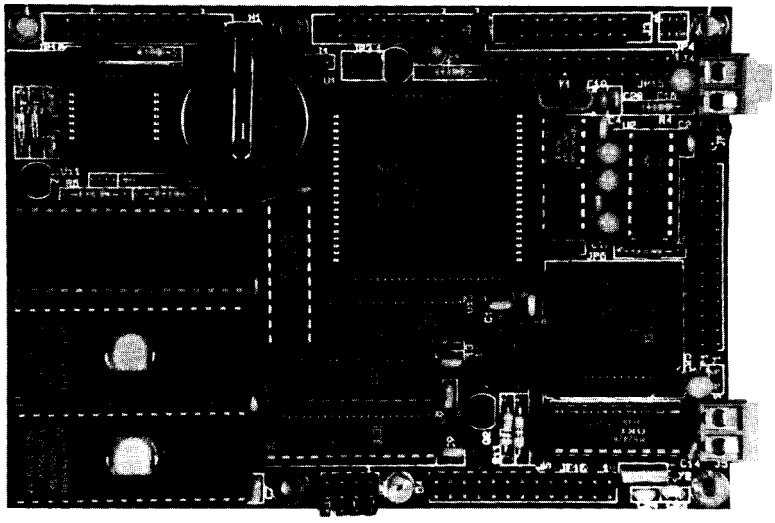
An EPROM emulator lets you avoid "Burn and Test" development cycles. In circuit emulation of 2K-32K 27xx series EPROMs. ABS Plastic case. Assembled or compiled code is downloaded directly to the target hardware.

**Cottage Resources** corporation
**Suite 151.10271 South 1300 East**
**Sandy, Utah 84094**
**VISA/MC,** COD. Call to Order: **(801) 268 - 2875**

Reader Service #128

# V25 Power Comes to Embedded Control!

Micromint's new RTCV25 is the perfect marriage of a PC-compatible processor, programming convenience, and control I/O. The heart of the RTCV25 is the NEC V25 microprocessor, an all-CMOS, 8088-compatible device running at 8 MHz. The 3.5" x 5" V25 offers engineers 16-bit processing power, large address space, and compatibility with many of the most popular and useful software development tools available today. The RTCV25 enhances the V25's power with 40 parallel I/O lines; 8-channel, 8-bit A/D conversion; two serial ports (one RS-232 and one RS-232/RS/485); up to 384K RAM and EPROM;a battery-backed clock/calendar; 1 K bit EEPROM,ROM monitor, and the RTC stacking bus. The RTCV25 is compatible with the full line of RTC peripheral boards and products.



Actual size
3.5" x 5

**Features**
- 8MHz V25 processor
- 2 Serial ports
- 40 Parallel I/O lines
- 8-channel, 8-bit ADC
- RTC Stacking Bus
- Small 3.5" x 5' size
- 5-volt only operation

**Options**
- 128 bytes EEPROM
- Battery-backed Clock
- 384K RAM and EPROM
- 8-channel, 10-bit ADC
- ROM Monitor

100 Quantity
OEM Configuration **$279.00**

**MICROMINT, INC.**
4 Park Street
Vernon, CT 06066
**call 1-800-635-3355**
**(203) 871-6170**
**Fax: (203) 872-2204**

Reader Service # 167

# Parallel Filters

## Design and Simulation by Computer

**PRACTICAL ALGORITHMS**

*Charles P. Boegli*

***The difficulties many have when hying to work with parallel filters has resulted in much of their capabilities left unexplored, but using them can be much easier if a computer does most of the dirty work.***

You can use a computer to design a parallel filter without laboriously setting up and testing every possible arrangement. Simulation of parallel-T filters by computer is particularly simple. The filters can be applied to data in records of indefinite length, or in real time using a previously developed algorithm [1], because they use only low- and high-pass filters. They achieve a range of characteristics you can conveniently explore using a spreadsheet program. I will discuss two such programs and show some of the filter characteristics they can simulate.

The parallel-T filter shown in Figure la is noted for its sharp rejection of a single frequency. In fact, the attenuation at the "resonant" frequency can be infinite with precisely sized components. By deliberately altering component values, you can reduce the severity of the attenuations.

This filter has a rather complex transfer function [2]. But the design of Figure la was used by analog-circuit designers before isolation amplifiers were easily available; they had to achieve filter characteristics with the least precise components possible. I will show you how simpler circuits (meaning those more easily simulated by computer) obtain practically the same characteristics. When these simple circuits are used, the parallel-T turns out to be an example of a number of filters having noteworthy properties.

### ANALYSIS

The response of a single low-pass filter section (Figure lb) is described by the Laplacian expression

$$G_L(s) = \frac{1}{(1 + Ts)} \qquad (1)$$

For two identical filters in cascade, separated by a unity-gain isolation amplifier [3], the expression is

$$G_L(s) = \frac{1}{(1 + Ts)^2} \qquad (2)$$

In these expressions $T$ is the time constant of the filter, which you can conveniently measure for sample data in terms of a number of samples (it equals $P/2\pi$ where $P$ is the period in samples/cycle at the "turnover" point). The filter described by equation (2) has a flat response for periods greater than $T$, while for shorter periods the response drops off at 12 dB per octave.

Similarly, a single high-pass section having the same time constant (Figure 1c) is described by

$$G_H(s) = \frac{Ts}{(1 + Ts)} \qquad (3)$$

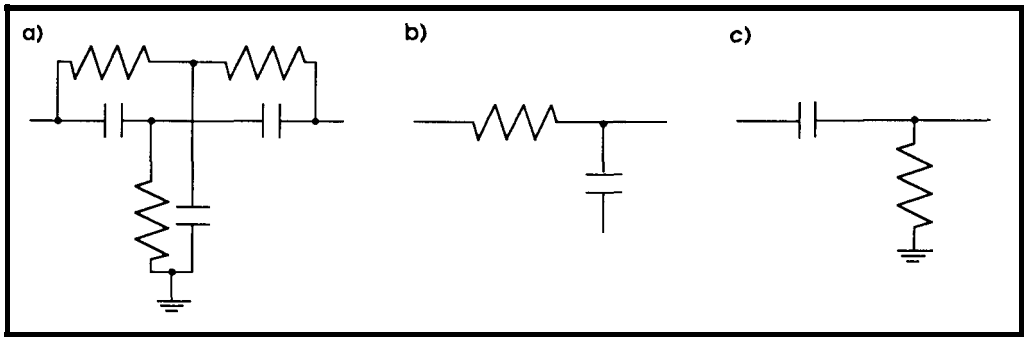



**Figure** 1 —*(a) An example of a parallel filter characterized by its sharp rejection of a single frequency and a complex transfer function. (b) An example of a single low-pass filter. (c)An example of a single high-pass filter.*

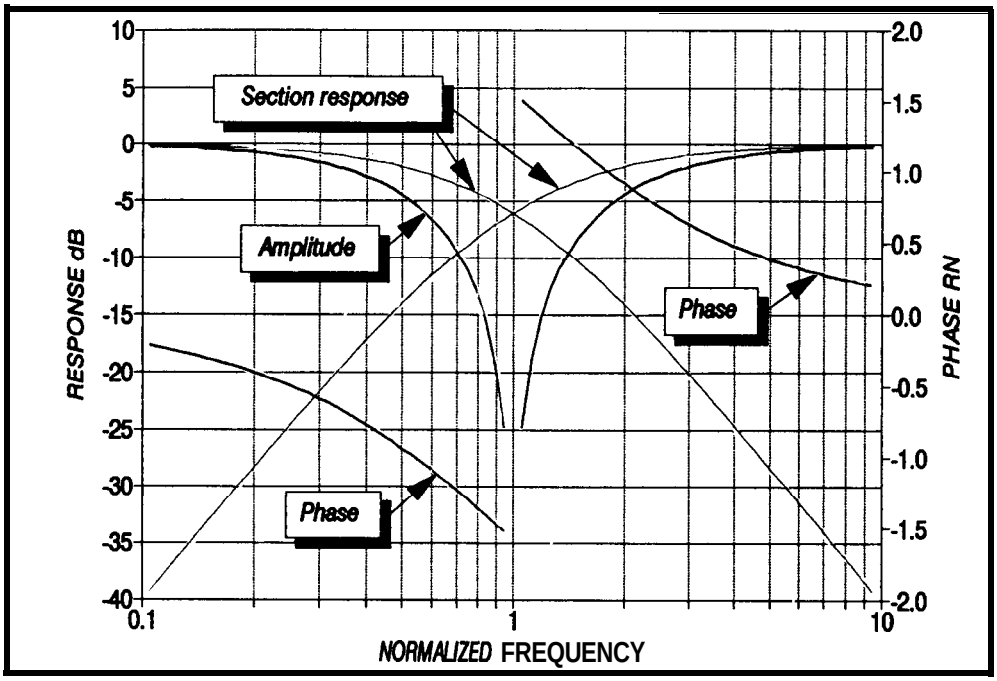



**Figure 2**—*The response of a simulated two-section parallel-T filter.*

and two isolated sections in cascade behave as

$$G_H(s) \quad \frac{(Ts)^2}{(1+Ts)^2} \qquad (4)$$

This filter's response is flat for periods less than $T$ and drops at 12 dB per octave for longer periods [4].

Assume a signal is split into two channels, one going through a two-section high-pass filter and the other through a two-section low-pass filter. After passing through the filters (all of which are assumed to have the same time constant) the two signals are added. The sum of the two signals has the transfer function

$$G(s) = \frac{\left(1+T^2s^2\right)}{(1+Ts)^2} \qquad (5)$$

You can obtain frequency and phase responses by substituting $j\omega$ for s and rationalizing the resulting expressions. "Normalize" equation (5) without any loss of generality by setting $T = 1$. The magnitude of the response is

$$M = \frac{\sqrt{\left(1-\omega^4\right)^2}}{\left(1+\omega^2\right)^2} \qquad (6)$$

and the phase is

$$\phi = \tan^{-1}\left(\frac{-2\omega}{1-\omega^2}\right) \qquad (7)$$

However, you may omit many of the cumbersome steps leading to equations (6) and (7) by using a spreadsheet. If the substitution of $j\omega$ for s is carried out separately for the low-pass and high-pass filters, addition of real and imaginary parts leads directly to the magnitude and phase of the entire filter.

At $\omega = 1$ (the frequency where all the corners are located) the numerator in equation (6) is zero and the magnitude of the response vanishes. Where $\omega < 1$ the phase is the arctangent of a negative quantity, while where $\omega > 1$ the arctangent is positive. At $\omega = 1$ the phase appears to go through an abrupt transition from $-\pi/2$ to $+\pi/2$. I've simulated these essential characteristics of a parallel-T filter by a simple arrangement of low- and high-pass filters. Figure 2 is a graph of the response.

Total rejection occurs when the responses of the low-pass and high-pass sections are equal and $\pi$ radians out of phase at one frequency. Use at least two sections of both types of filter because each one has a phase shift of $x/2$ only at a period of 0 or $\infty$, depending on the type of filter. However, the number of sections has no upper limit because a shift of $\pi/2$ always occurs within a reasonable distance of the corner for every multisection filter.

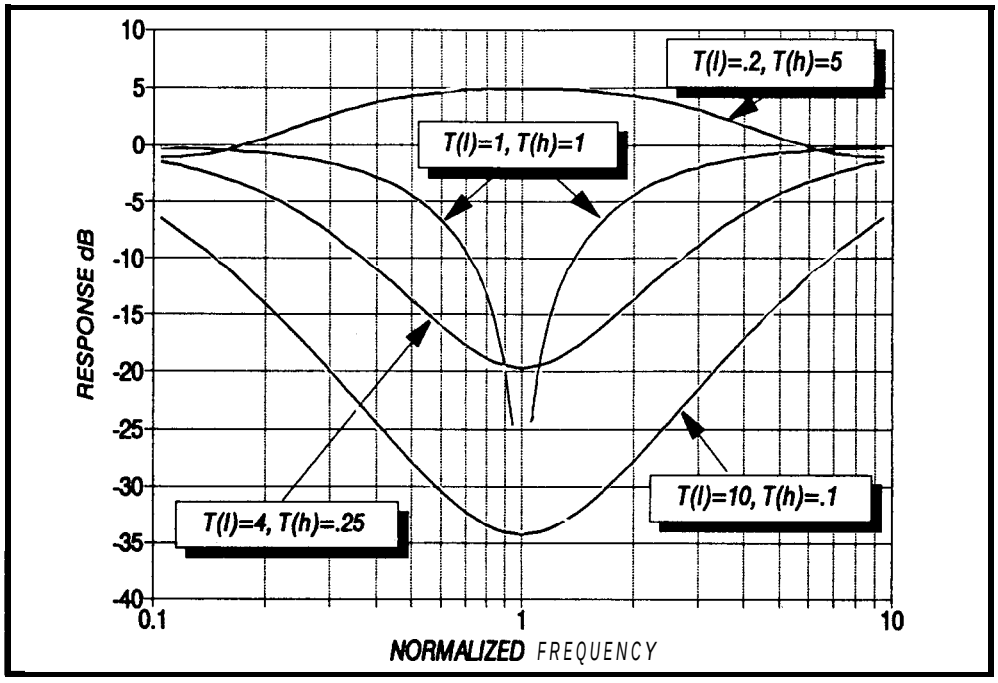



**Figure J-The** *responses of some two-parallel section fitters where the time constants of the high-pass sections are the reciprocal of those of the low-pass to maintain the center frequency of 1.0.*

### Improvement in Sampled-Signal Filter Algorithm

**Almost all** algorithms **for processing sampled signals contain approximations that produce errors in their** outputs. **An important task** in devising an algorithm is to reduce the errors as much as possible.

**In determining the time response of a parallel** filter at the **null frequency to obtain Figure 9. I discovered an** easily improved approximation in the algorithm published In my article "Filtering Sampled Signals" in issue #21 of CIRCUIT CELLAR INK (6). This determination is a **sensitive** test for residual errors because the response of such a filter is identically zero at the rejection frequency. The calculated magnitude of this response is consequently a direct indication of the algorithm error.

The error was greatly reduced by a simple **expedient** that materially increases neither the complexity of the algorithm nor the memory required for its **application.**

I improved low-pass filtering by using the arithmetic mean of the input samples at times $t$ and $t + 1$ (instead of that at $t + 1$ only) to find the filtered signal at $t + 1$. **Thus, the filtered signal at $t + 1$ is** $\exp(-1/T)$ **that at $t$ ($T$ being the filter time constant in samples/m) plus** $(1-\exp(-1/T))$ **times the mean of the samples at tand $t + 1$.**

High-pass filtering requires the added step of **subtracting** each value of a low-pass filter from the **input sample at $t+1$.**

**in the time** response at the null frequency of a **two-section** parallel filter with a sinusoidal input signal of unit peak amplitude, the earlier algorithm showed a **residual** amplitude of 0.0842 peak **with 50** samples per cycle. The improved algorithm reduces it to 0.0026.

If the effective "energy" input from one sample to the next **Is** the integral of the waveform between these samples, it is evident that the actual quantity to be used is the integral of the waveform times the sample interval (defined as 1.0). The actual value of this **Integral** cannot be determined because the waveform is unknown. The mean of the Iwo sample magnitudes appears to be a better approximation to this value than either sample alone, at least for sinusoidal waves.

I Incorporated this improvement Into the **spreadsheet** TIME-2.WQ1.

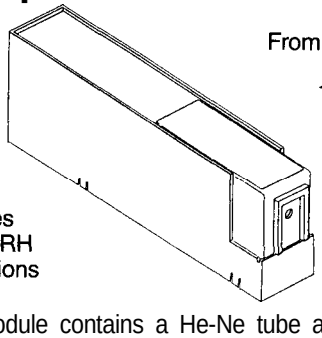
## SPREADSHEET PROGRAMS

Two-section filters

Filters consisting of two parallel two-section filters will be called "two-section filters" for the rest of the article. You can set each of the four time constants of such a filter independently using spreadsheet `PARAL-2.WQ1`. The gain of the long-period sections can also be varied, which in itself covers the complete spectrum of designs; altering the gain of other sections has no point because it


**LASERS**

**Complete Helium Neon Laser**



From **$59.00**

Complies with CDRH Regulations

This Module contains a He-Ne tube and matching power supply, all in one housing! Comes with instructions and a 1 year warranty. Makes an ideal Science Fair Project. Operates on 12 VDC @ 600 mA. Output: Visible Red 633nm Continuous.

0.5 **mW** Power Output ***Cat.*** *#HNKD-10* **$59.00**

*1.5* **mW** Power Output. 3 times the power as above! *# H N K L* **$75.00**

TOLL FREE ORDER # 1-800-722-0392

*FREE CATALOG*

*on these* and *other related items.*

**MEREDITH INSTRUMENTS**

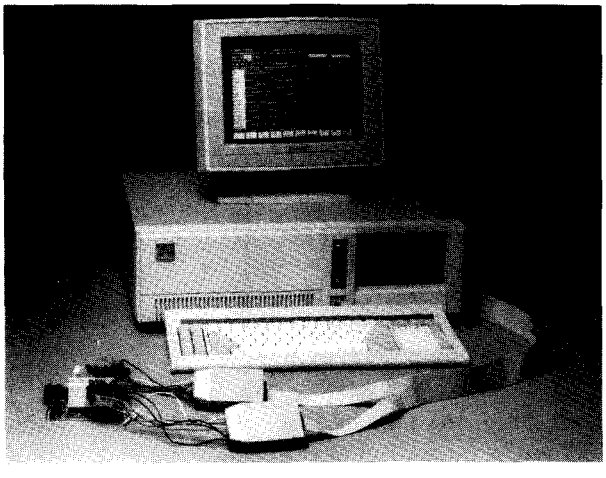
5035 N. 55th AVE., #5 / P. 0. BOX 1724 / GLENDALE, AZ 85301

Technical Service Line: **602-934-9387**

FAX Line: **602-934-9482**

Reader Service #161

**PC-Based Logic Analyzers**



**Special Anniversary Sale !!**

ID160 for $495 (regular $695)
ID161 for $595 (regular $895)
ID320 for $995 (regular $1595)

'50 MHz or 100 MHz Sampling ● 8K Trace Buffer • 16 or 32 hannels *Multi-Level Triggering *State Pass Counting 'Event Timer/Counter *Performance Histograms *Hardcopy )utput *Disassembles popular 8-bit micros *and much more

.imited Time Offer. Money Back Guarantee

INNOTEC DESIGN, INC.
6910 Oslo Circle, Suite 207
Buena Park, CA 90621
Tel: 714-522-1469 FAX:714-527-1812

Reader Service #152

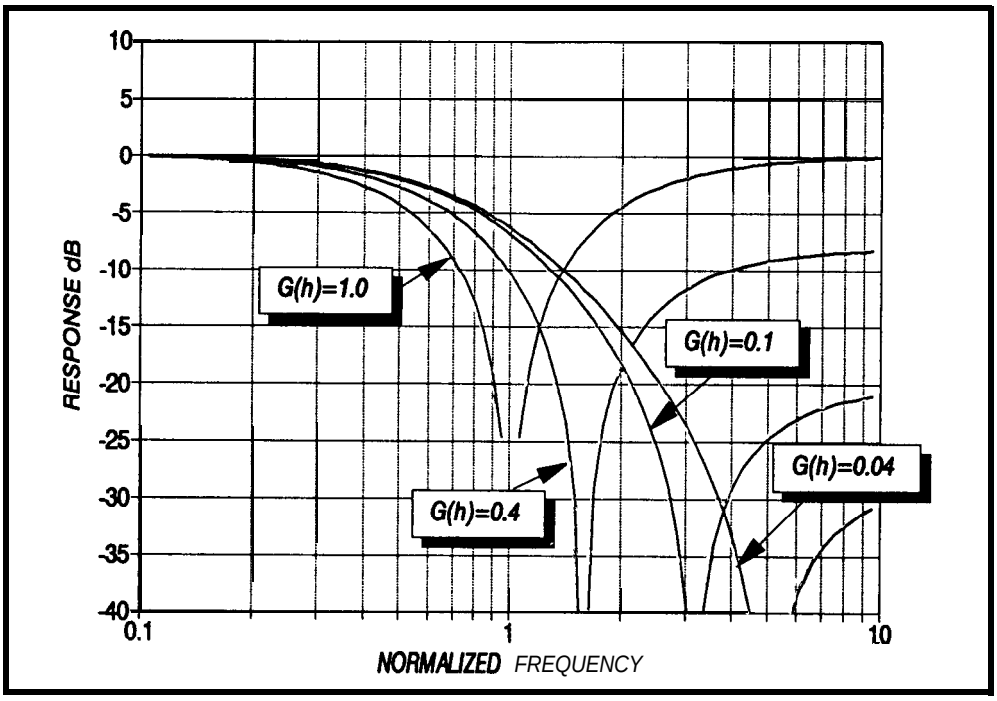



Figure 4—*The response of two-section parallel filters with various high-pass gains in which the total rejection of a single frequency apparently still occurs.*

can be greater or less than 1 .O. *[Editor's Note: Software for this article is available from the Circuit Cellar BBS and on Software On Disk #24. See page 105 for downloading and ordering information.]*

If the time constants of the low- and high-pass sections are different, total rejection of one frequency does not occur. The response continues to exhibit a dip as long as the frequency of the low-pass filter is greater than that of the high. The opposite condition produces a broad peak in the response.. Figure 3 shows typical curves for cases in which the time constants of the high-pass sections are the reciprocal of those of the low-pass to maintain the center frequency of 1.0. However, if the time constants of all the filters are identical, but the gain of one set differs from that of the other, total rejection of one frequency apparently occurs. The rejected frequency merely moves off from $f = 1.0$ as shown in Figure 4. This parallel-T filter yields a sharper cutoff than an assembly of simple filters, but at the expense of greater gain at a distance.

Also, the spreadsheet can find the response that results from subtracting (instead of adding) the outputs of the low- and high-pass filters. The use of a negative gain for the high-pass filters is necessary. Typical curves are shown in Figure 5.

Three-section filters

The spreadsheet PARAL-3.WQ1 is designed similarly for parallel arrays of three low-pass and three high-pass filters. All six time constants can be set independently. You may use the spreadsheet for two-section filters if the time constant of one of the high-pass filters is made very large or one of the low-pass filters very small. Figure 6 shows responses, analogous to those of Figure 2, for a three-section filter in which all sections have a gain of 1.0 and the time constants of the low-pass sections are reciprocal to those of the high-pass sections.

The steeper cutoffs of the three-section filter somewhat sharpen the null in the parallel filter, which can be seen in Figures 2 and 6. The phase shift of three identical cascaded high-pass sections is not n / 2 at $\omega = 1.0$; the time constants must be $3^{+0.5}$ (and that of the low-pass sections $3^{-0.5}$) to get total rejection at $\omega = 1$ .O. When the high-pass part's gain is altered, the response no longer goes through a complete null; in this respect, the three-section filter is not as "well behaved" as the two.

## SIMULATION IN TIME

Frequency-domain simulation of parallel filters allows rapid, easy observation of their responses and the

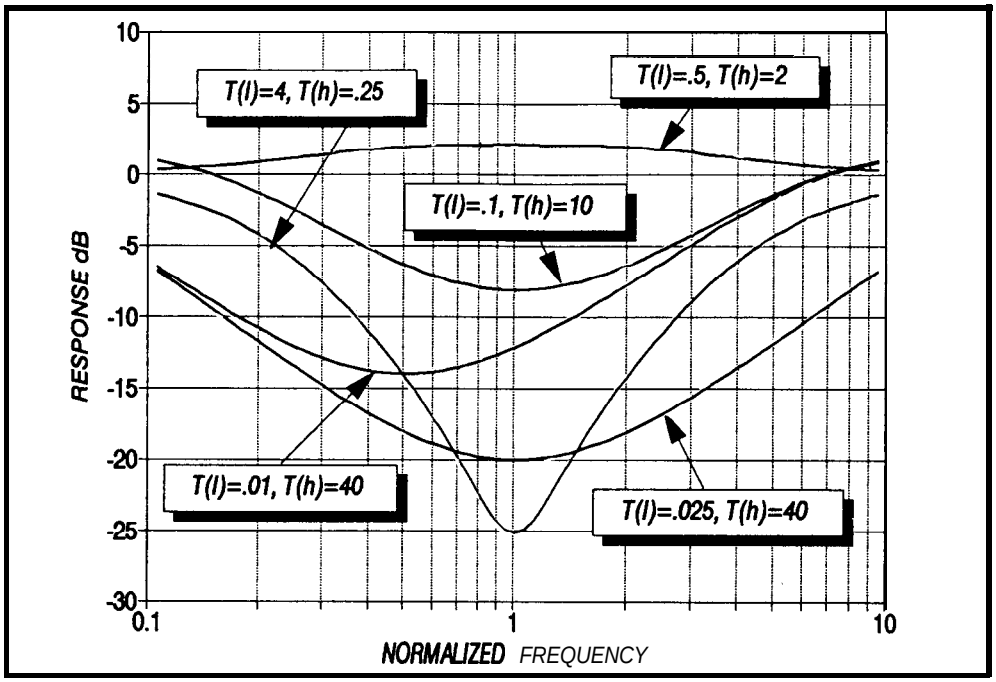



**Figure 5**—*The response that results from subtracting the outputs of low- and high-pass filters.*

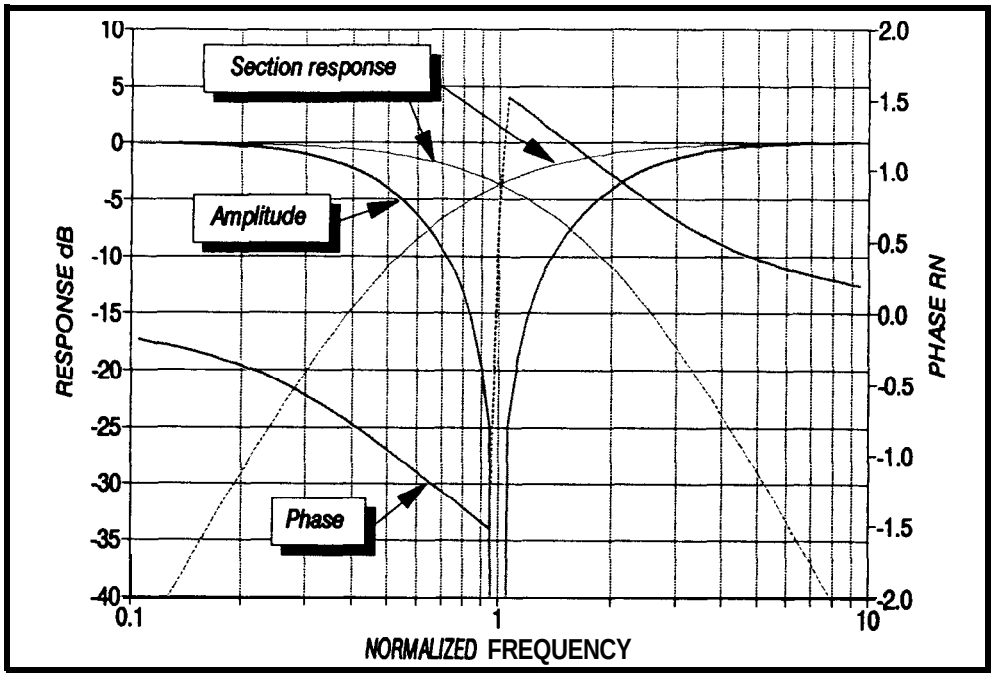



**Figure 6**—*The responses for a three-section filter in which all sections have a gain of 1.0 and the time constants of the low-pass sections are reciprocal to those of the high-pass sections.*

effects of manipulating the constants. You can also simulate these filters in the time domain using a spreadsheet if an algorithm is available to perform the necessary filtering. In my article "Filtering Sampled Signals" in issue #21 of CIRCUIT CELLAR INK, I showed how to apply low- and high-pass filters in the time domain to sampled signals, and I developed algorithms that can be readily applied directly to parallel filters [5].

An implicit advantage of this kind of simulation is cumbersome calculations for frequency-domain simulation are unnecessary. However, certain disadvantages exist. The sampling frequency must be high enough compared to the signal being sampled that certain approximations will not unduly degrade the results. Also, you can only examine the parallel filter's performance a frequency at a time.

Figure 7 is a time response drawn with the spreadsheet TIME-2.WQ1. I subjected the incoming wave to two successive low-pass and high-pass filterings simultaneously, and then merely added the results sample by sample. The time constants of all the filters in this figure are the same as the radial period of the incoming signal, so the output signal is zero. When


**The $595 Solution**
**to 8051 System Development**

**PDK51**



The PDK51 is a fully integrated hardware, firmware, and software system designed to help you develop your products quickly and cost effectively.

All you need to use the PDK51 is an IBM-PC/XT/AT or compatible. We supply the rest.

- SIBEC-II microcontroller board (8052AH-BASIC CPU) with 16K RAM
- BASIC interpreter source on disk
- SIBEC-II hardware manual with schematic
- Monitor/debugger ROM plus manual
- Monitor/debugger hardcopy source listing
- UL listed 5V power supply
- Programming power supply adjustable from 4V to 26V
- KERMIT communications program
- SXA51 8051/8052 cross assembler
- BTK52 BASIC programmers toolkit with tutorial
- 160 page BASIC Programming Manual by Systronics
- RS232 cable
- PDK51 tutorial
- Optional: aluminum case—$45, monitor/debugger source—$25, battery-backed real-time clock—$39.
- All components are manufactured to exacting standards and warranteed for one year.

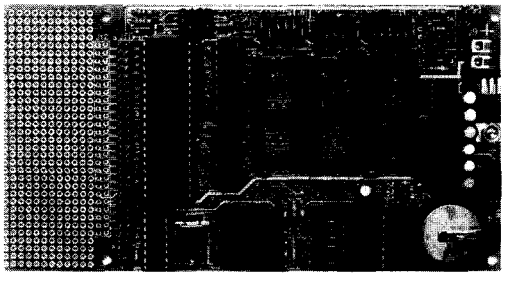
**PDK51** PLUS includes everything in the PDK51 plus Vers. 3 of our popular BXC51 8051/8052 BASIC compiler-$800.

Call Now! 603-469-3232 or FAX 603-469-3530

Binary Technology, Inc.
Main Street . PO Box 67 . Meriden, NH 03770

Reader Service XI 14

**EMBEDDED CONTROLLER**
***for Quick Development***



The EC-32™ is a versatile 80C32 microcontroller board. It is ideal for quickly developing products, prototypes or test fixtures.

- 80C32 microcontroller (8051 compatible)
- BASIC-52 or MONITOR-52 available
- Program in C, BASIC or assembly language
- 8 to 92K RAM, EPROM or EEPROM
- Breadboard area and expansion bus
- RS-232 port and 12 digital I/O lines
- $100 for 11 MHz, $145 for 20 MHz

**Iota Systems, Inc.**
(702) 831-6302
POB 8987
Incline Village, NV 89450

**20 MHz Board Available**

Reader Service #153

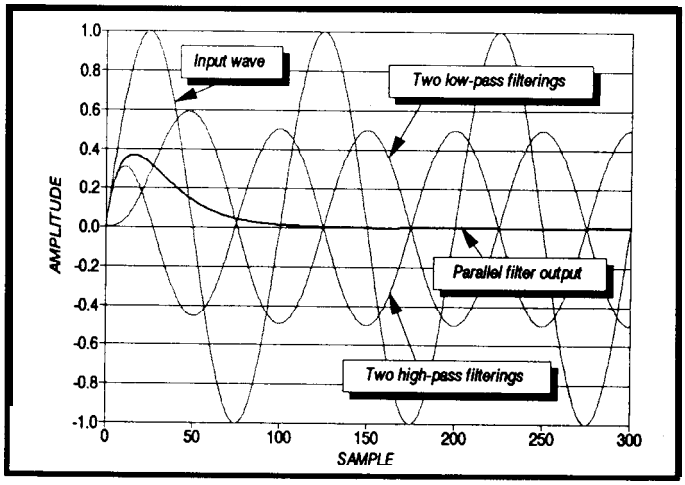



**Figure 7**—*The time behavior of a parallel filter using spreadshet TIME-2. WQ1 at 100 samples/cycles sampling rate. The incomin wave is simultaneously subjected to two successive low-pass an high-pass fitters. and the results are added sample by sample.*

the period of the incoming signal is changed, the output is no longer zero, which the frequency-response graphs show.

The starting transient seen in Figure 7 appears in the output record until the response has had time to settle. You can eliminate the response by applying a suitable Hanning time window to the incoming data or just ignoring it until it settles.

**REFERENCES**


1. Charles P. Boegli, 'Filtering Sampled Signals,' CIRCUIT CELLAR INK June/July 1991, pp. 98-101.
2. This filter is extensively discussed in James, Nichols. and Phillips, *Theory of Servomechanisms (New* York, McGraw-Hill Book Company, 1947). p. 117ff. lhe complexity of the transfer function arises from mutual loading effects.
3. The isolation amplifier prevents the second section from loading the first. The second section Is also presumed to be followed by an isolation amplifier to avoid loading.
4. The responses of both low and high-pass sections do not actually show a sharp transition at the 'corner' frequency, but rather change smoothly from one slope to the next. A single-section filter of either type has an attenuation of 3 dB at the 'corner.'
5. Charles P. Boegli, loc. cit. Also see the sidebar.
6. Charles P. Boegli, loc. cit.



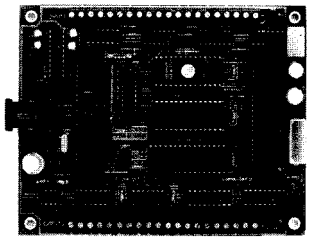
*Charles Boegli is president of Randen Corporation* in *Blanchester, Ohio. Randen is a small company offering* services *in technical computer programming and analog circuit design.*


## IRS

425 Very Useful
426 Moderately Useful
427 Not Useful


**DG32F** *8031* Controller Board with FLASH MEMORY

***Eliminates the need for Programmer and UV Eraser***



- 8031 processor
- 11.0592 MHz clock
- 32 K Flash Memory
- 16 K EPROM
- 8 K RAM capacity
- 16 Buffered Input Lines
- 16 Buffered Output Lines
- RS-232D Serial Port
- Serial Peripherial Interface Port
- Supply Voltage Supervisor Circuit Board size 4.8" x 6"
- EPROM monitor with Flash management program
- Application software

**A&T BOARD $155** - PARTIAL KIT (all components and documentation except Flash Memory & screw terminals) $1 10 PCB - bare board with documentation (manual, shematic and parts list) $ 40

L.S. ELECTRONIC SYSTEMS DESIGN
2280 Camilla Rd. Mississauga, Ontario, Canada
L5A 2J8 Phone/Fax: (416) 277-4893

Terms: Shipping US/Canada $6 Check or Money Order, Please

**HAJAR ASSOCIATES**
**NATIONAL ADVERTISING REPRESENTATIVES**

**NORTHEAST**
**Debra Andersen**
**49** Walpole Street
Norwood, MA 02062
(6 17) 769-8950
Fax: (6 17) 769-8982

**MID-ATLANTIC**
**Barbara Best**
**569** River Road
Fair Haven, NJ 07704
(908) 741-7744
Fax: (908) 74 l-6823

**SOUTH EAST**
**Christa Collins**
**7640** Farragut Street
Hollywood, FL 33024
(305) 966-3939
Fax: (305) 9858457

**MIDWEST**
**Nanette Traetow**
**242** East Ogden Avenue,
Suite A
Hinsdale. IL 6052 1
(708) 789-3080
Fax: (708) 789-3082

**WEST COAST**
**Barbara Jones**
**& Shelley Rainey**
**3303** Harbor Blvd.,
Suite G-l 1
Costa Mesa, CA 92626
(7 14) 540-3554
Fax: (7 14) 540-7 103

**RESERVE YOUR AD SPACE TODAY!**

# DOMESTIC AUTOMATION

# Home Automation: Read All About It

*Ken Davidson*

We often have people new to home automation call us or leave us a message on the Circuit Cellar BBS asking how to find out more.

"Are there any good books on the subject?"

"I'm building a new home and want to prewire it for home automation. What should I do?"

Up to now, I've always had to go into a song and dance about how the industry is still young, there really isn't much out there for the novice, and most systems on the market are specialized enough that it would be difficult to write a general-purpose book about home automation.

I'm happy to report that I can finally stop dancing (I'm awful anyway; just ask my wife) and start recommending some real books. I came across two recently and want to give you my impression of them.

## HOW TO AUTOMATE YOUR HOME

Written with the novice in mind, *How to Automate Your Home* does an excellent job of covering a broad range of subjects dealing with all aspects of home automation. Author David Gaddis starts with an overview of the major players in the industry, including X-10, Smarthouse, CEBus, and Echelon. Because the latter three were in a state of flux at the time the book was written (as, for all intents and purposes, they still are), some of the details he presents about them were dated even before the book was published. (Now you know why nobody has been anxious to publish a book before now.) Of the group, X-10 is the only system actually on the market that the individual can obtain, so he uses it as the basis for much of the book.

Gaddis continues with an overview of the various media used in home automation, and spends a chapter talking about wiring homes, both new and retrofits. From there, he spends several chapters discussing home security, appliance and lighting control, audio/video systems, telephones, and environmental control systems. In each case, he presents equipment currently on the market (and does a good job of covering more than just X-10), makes suggestions for how to plan your own automation setup, and presents installation examples that you can use as the basis for your own system.

The information presented is very good, and Gaddis does a wonderful job of covering virtually everything available on the market, but the presentation itself is terrible. The editing is awful, and the book appears to have been prepared using just a word processor (no interesting twists to the layout that can only be provided by page layout software). It also looks like it was printed on a dot matrix (or a bad inkjet) printer. The least he could have done was beg, borrow, or rent a laser printer for the final output. Coupled with cheap paper and poor printing, the bad text makes spending much time reading before eye fatigue sets in difficult.

***"I'm building a new home and want to prewire if for home automation. What should I do?"***

Most of the figures appear to be bad copies from many different sources. I even recognized some lifted directly from copyrighted material (including the book I'm going to talk about next that I know was on the market first) with no attribution given.

The book also lacks a much-needed index, which might help lessen eye fatigue because you wouldn't have to search as long to find a particular subject.

All in all, *How to Automate Your Home* is an good introduction to the world of home automation if you can get past the poor publishing job. The book runs 128 pages, includes 115 illustrations, and is available for $29.95.

**Home Automation, USA**
**P.O. Box 22536**
**Oklahoma City, OK 73123**
**(405) 840-4751**
**Fax: (405) 842-3419**

## AN INSTALLER'S GUIDE TO CEBUS HOME AUTOMATION

Let me start by saying this book is most certainly not for everyone. Written by Michael Maahs in conjunction with Parks Associates, *An Installer's Guide to CEBus Home Automation* is aimed squarely at professional installers (electricians, alarm installers, etc.) who need a practical, down-to-earth description of how wiring should be installed in new (and, in some cases, existing) construction to accommodate present and future home automation needs. While most of the book focuses on CEBus, much of the material applies equally to other systems on the market.

The book begins with an introduction to CEBus and home automation. Then it describes where the outlets (power, telephone, and coax) should be installed in each type of room (living room, kitchen, bedroom, etc.), and why you might want more than what is usually installed in a traditional nonautomated house. It also explains the reasons for some of the nontraditional placement of outlets (e.g., a power outlet up near the top of a window to accommodate remote-controlled power drapes).

After talking about placement, Maahs then discusses how and in what order the various media should be run. Separate chapters discuss wiring a central node panel, the use of "66" blocks, and running power lines, twisted pairs, and coax cables. Later chapters provide guidelines for estimating material amounts and costs, and make specific recommendations for tools required. Finally, the author makes a brief foray into advanced A/V installations.

The presentation is very nice, with the information organized in a clean, readable format. All illustrations were drawn by Michael Cogbill on a computer specifically for this book and are well done. *An Installer's Guide to CEBus Home Automation* is 90 pages long, includes an index, and is spiral bound so it sits flat on the table without argument. The price on the book ($149) will probably drive many people away, but if you're building a house and want to prewire so you're ready for the future, I think you'll find it worth the price. Certainly, any current or future professional installers are foolish not to own a copy.

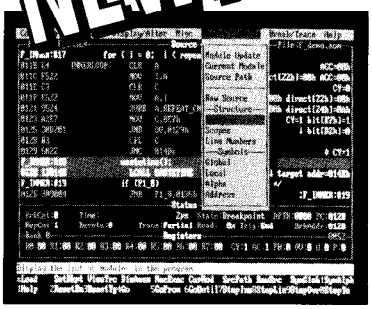
**Parks Associates**
**5046 Radbrook Rd.**
**Dallas, TX 75220**
**(214) 490-I 113**
**Fax: (2 14) 490- 1133**

*Ken Davidson is the managing editor and a member of the Circuit Cellar INK engineering staff. He holds a B.S. in computer engineering and an M.S. in computer science from Rensselaer Polytechnic Institute.*

IRS

**428** Very Useful
429 Moderately Useful
430 Not Useful

NEW! iceMASTER™
COP8 8051 68HC11



YOUR WINDOW TO EMULATION PRODUCTIVITY

- Easy to learn & use
- Windowed interface -- user configurable
- **FAST! Download -- < 3 sec. typ. at 115KB**
- Source Level debug
- **A 4K frame trace buffer** with advanced searching capabilities.
- **Hyperlinked On-line help** guides you through the emulation process.
- **iceMASTER** connects easily to your PC, requires no disassembly, or expansion slots. Works on any PC (DOS or OS/2), MicroChannel or EISA. Even laptops!
- **Supports more than 50 different 8051 family derivatives.** M68HC11 support will be available early in 1991.
- **Try iceMASTER risk free!** Satisfaction Guaranteed or return for a full refund!*
- RENTALS AVAILABLE! Ideal for consultants and researchers!
- Call today for **free demo disk and ask about a free 8051 Macro** Assembler! **(800) 638-2423**

MetaLink® Corporation

FREE ASSEMBLER

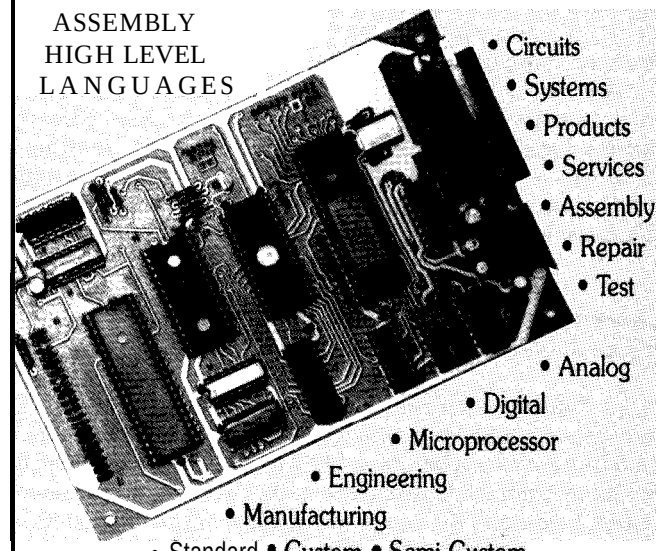
MetaLink Corporation **PO Box 1329** Chandler, Az 85244-1329
**Phone-** (602) **9260797** FAX (602) 926-1198 TELEX: 4998050MTLNK
* Wii 10-day trial period

Reader Service #163

MAGNUM OPUS
CUSTOM ELECTRONIC ENGINEERING

SINGLE BOARD COMPUTERS
EPROM PROGRAMMERS

ASSEMBLY
HIGH LEVEL
LANGUAGES



• Circuits
• Systems
• Products
• Services
• Assembly
• Repair
• Test
• Analog
• Digital
• Microprocessor
• Engineering
• Manufacturing
• Standard • Custom • Semi-Custom

284 **Kennedy St • Iselin, NJ** 08830
(908) 283-4925

Reader Service #158

# CONNEC-TIME

*Conducted by*
*Ken Davidson*

## *Excerpts from the Circuit Cellar BBS*

**The Circuit Cellar** BBS
300/1200/2400 bps
24 hours/7 days a week
(203) 871-1988
Four Incoming Lines
Vernon, Connecticut

*The text-only environment of a BBS doesn't lend itself to doing graphics, but some adven turous souls often t y their hand at "ASCII art." The first thread, dealing with NiCd batteries, has a schematic that I redrew to make easier to read. The second discussion on using RJ-11 connectors for serial connections has some art left intact.*

*Put a bunch of engineers on a BBS, bring up the topic of rechargeable batteries, and you'll end up with enoughgood information fo-write the definitive NiCd book. Read on...*

**Msg#:46149**
From: RICHARD FOLEA To: ALL USERS

Can anyone tell me what I did wrong? I have a Panasonic camcorder (one of the large ones that you rest on your shoulder; I can't remember the model number) that uses a larger rechargeable battery; 12 volt, I think. After about three months of use, the battery wouldn't take a charge anymore. That is, the charger would "charge" the battery, the indicator saying the battery was charging would go off, and I would put the battery in the camera. The battery would only last a few minutes.

Is there something goofy about these batteries that you can't leave them in the charger all the time, or should I occasionally suck all of the current out of it and fully recharge it? Now that I have a battery that acts like this, can I fix it by shorting the terminals with a small resistance? I'd appreciate any suggestions. Also, does anyone know where I can get new batteries? The local mall wants $50! There has to be a better way.

**Msg#:46151**
From: RUSS REISS To: RICHARD FOLEA

Yup! Right on both counts. NiCds (which these undoubtedly are) get unhappy both with (SOME) continuous trickle charge, and incomplete cycling. You MIGHT be able to bring them back to life by completely discharging, then recharging a number of times. I wouldn't discharge too quickly, or the internal heating might cause damage, but some resistance that approximates normal usage, I'd guess.

Try calling Periphex Inc. at (800) 634-8132. The fellow there (I forget his name) is very helpful, sells inexpensive packs, and I'm sure can give you some tips and get you rolling again. Note: "inexpensive" NiCds are never inexpensive though. You can save SOME $$ from him, but they won't be CHEAP. I buy my video recorder and ham radio battery packs from him, though, and they are very reliable.

**Msg#:46154**
From: TIMOTHY TAYLOR To: RICHARD FOLEA

For batteries, try MCM Electronics, (800) 543-4330. I've purchased a number of batteries for my Sony camcorder. One thing to remember is to completely discharge the batteries before the charge cycle begins. That'11 somewhat alleviate the "memory" problem. Also, follow manufacturer directions completely when you get new NiCds. You'll really extend the life of 'em.

**Msg#:46164**
From: PHIL COVINGTON To: RICHARD FOLEA

My Panasonic camcorder uses a 12-V gel lead-acid battery; the number on the battery is PV-BP80. I bought an extra one recently and the instructions that came with it states that you should *NOT* discharge the battery completely and should charge the battery immediately after use. These directions seem to be consistent with the battery type.

**Msg#:46175**
From: RUSS REISS To: PHIL COVINGTON

You are right! Lead-acids should never be allowed to "drain dry" and should not be stored in a dead state. Gee, I've not seen gel cells in camcorders. I've always liked them better than NiCds in many respects.

**Msg#:46207**
From: PHIL COVINGTON To: RUSS REISS

I was expecting to find NiCds too, until I opened an old battery from my camcorder and found that it was a lead-acid type-it has six cells with rubber vent caps. The manual for the camcorder also had a warning about breaking the battery and coming in contact with the electrolyte containing sulfuric acid.

Radio Shack's 1992 catalog has a replacement battery that appears to be identical to the battery I have, but the catalog states that it's a NiCd battery pack. Having recently purchased an extra battery pack for my camcorder, I am now wondering if it is a NiCd or lead-acid type. The original Panasonic battery pack has a thin cover that can be pried up and under it you can see the rubber caps for each of the six cells; the new battery (not a Panasonic) appears to be completely sealed. If some of these replacement battery packs are in fact NiCds, I wonder what effect charging these packs with a charger designed for the original lead-acid battery has on the life of the new packs?

Do gel cells suffer from the same problems of incomplete cycling as NiCds? I was wondering if anyone makes a gel cell replacement battery pack for my 2-meter HT.

**Msg#:46249**
From: STEVE CIARCIA To: PHIL COVINGTON

Generally speaking if your device comes with a modular transformer and not one of those 1-3 hour intelligent outboard chargers, you can make the following assumption: if the charging instructions say plug in the charger overnight, or leave the charger plugged in all the time, but nothing about charging it too long, it is probably NiCd. If the instructions say plug it in overnight but no longer, or something like "Do Not Overcharge," then it is probably lead-acid. NiCds can trickle charge but lead-acids don't like it after they are at a certain level.

You charge a NiCd with a constant-current charger and a lead-acid with a constant-voltage charger. The little modular supplies described above are more like current chargers and therefore only suitable up to a certain point on a lead-acid battery.

**Msg#:46250**
From: RUSS REISS To: PHIL COVINGTON

Well, Steve pretty much answered it Phil. As for gel cells for HTs, no, I've never seen any. That Periphex number I gave earlier in this thread is run by a ham, mostly for hams. I have thought about making a "battery belt" one could wear that would provide long-term power for HTs for emergency communications, camping, hiking, biking, sailing, and so forth. Never did go far with the idea, though.

**Msg#:46186**
From: ANDY SARNAT To: RUSS REISS

I couldn't help feeling a little deja vu when I read the original question about the failing NiCds: the battery pack in my Toshiba laptop had exactly the same symptoms after about 6 months (i.e., a full "charge" only lasted a few minutes). But I already knew about the NiCd memory problem and I thought I treated that battery pack real nice from the start. I always deep discharged it, I never overcharged it or left the charger plugged in overnight, did everything but take it to dinner and buy it flowers, and this is the thanks I get! So what went wrong?

I even tried to rejuvenate it by fully recharging (until the LED indicated "full"), then discharging to zero by leaving the screen backlight on full intensity until dead, then repeating the cycle six or seven times. No dice.

The local Toshiba dealer says "everybody has to replace their battery pack once a year or so," but of course he SELLS batteries. So I shelled out for a new battery pack, then pried open the old pack to find six NiCd cells all connected in series with metal straps spot-welded to their cases, so I don't see any prospect for replacing the cells myself, even though I've been told that probably just one of the cells is bad.

So I've got three questions for the battery gurus:

(1) Whuddya do to prevent this?
(2) Is there a practical way to replace welded-together cells?
(3) How come my Dustbuster sits on my wall trickle charging continuously for years, hardly ever gets deep discharged, and is still as strong as a Clydesdale?

**Msg#:46205**
From: BOB ARMSTRONG To: ANDY SARNAT

You can replace those NiCd cells just by soldering in a replacement for the one (or more) that's gone bad. Just rough up the surface some with a file or sandpaper to make it take solder, use some solder wick as a strap, and off you go. I do it all the time. Just don't short the cells... they get kinda hot!

**Msg#:46217**
From: ANDY SARNAT To: BOB ARMSTRONG

Thanks, I'll give it a shot. One more question: can you mix different size NiCd cells in series? Since each cell handles the same current, would the smaller ones suffer damage in series with larger ones under normal load? The reason I ask is that the cells in the battery pack don't seem to be a standard size. They are cylinders about 5/8" diameter and 2.5" long. What would happen if I substituted AA cells for one or more of them?

**Msg#:46219**
From: JEFF BACHIOCHI To: ANDY SARNAT

You can mix sizes, but current (when the cells are in series) will be limited to the maximum of the smallest cell.

Check Digi-Key. They offer battery packs and individual cells. Sizes are NOT limited to the standard AAA, AA, C, and D, but include 1/2's, 1/3's, 2/3's, and other oddities!

**Msg#:46248**
From: STEVE CIARCIA To: ANDY SARNAT

Different size NiCd cells have different charge (and trickle charge) ratings. The 100-mA charge on a D-cell will fry an AA cell. Be careful about radically different sizes in series.

**Msg#:46252**
From: BOB ARMSTRONG To: ANDY SARNAT

They sound like "sub-C" cells. Check your local hobby store for cells used in model cars and airplanes. They range from 1000 to 1800 mA and can be purchased individually or as a 6- to 8-cell pack. Packs average $25 to $40 each.

**Msg#:46294**
From: JAKE MENDELSSOHN To: ANDY SARNAT

Do not ever try to deep discharge a NiCd by just leaving the equipment on and draining it to zero. You can get a situation called voltage reversal where the voltage of the NiCd is actually switched. Once that happens, you will not be able to recharge the NiCd.

Here is what happens to NiCds in series: Assume you have five NiCds in series, each with a full voltage of 1.2 V (carbon-zinc batteries have 1.5 volts per cell; NiCds general have only 1.2 V). The five new NiCds in series give you 6.0 volts. Now after using this NiCd set for a while, one of the cells develops an internal short (see message 46157). This cell is shorted out and is no longer in the circuit. Now when you try to recharge the set as you did before, there are only four cells in the series circuit. Your charging circuit will try to charge them to a total of 6 volts as before. This forces each cell to 1.5 volts per cell. This overcharging damages the other previously OK cells. The end result is you now have one cell damaged with internal shorts and four cells damaged by overcharging. The whole set is destroyed and there is not much you can do, except as the salesman said, "Everyone has to replace their battery pack once a year or so." The 500 recharging cycles claimed in the advertisements is a theoretical limit that I have never even come close to.

If you use a very sharp wire cutter or a Dremel tool you can cut the thin metal connectors that are welded between the different cells. Then try soldering your new good cell into the series circuit. The new cell you are adding should have the same level of charge as the other cells in the series or you end up with the same uneven charging problem. I always charge all cells individually to 100% before I solder them together and use them as a team.

There are different configurations of cells with different chemicals, different chemical ratios, and different internal structures. Some cells are meant to use used in situations where they are constantly cycled up and down, others are normally in a fully charged standby mode and supply current only rarely. Some cells are meant to supply heavy current for a short time, others low current for long periods. You really need to match the specific type of cell for the specific use you have in mind.

Gel cells do not seem to suffer from the same "memory" and "internal short" problems that affect NiCds.

The amp-hour rating of these cells is based on a 20-hour discharge rate. Thus, a IO-AH cell will give 0.5 amp for 20 hours (0.5 x 20 = 10). You will not be able to get 10 amps for 1 hour out of the cell. The optimal recharging rate for a cell is 1 /10 to 1/20 of the load rating. Thus you should recharge a IO-AH cell at a 1- to 0.5-amp rate.

**Msg#:46157**
From: JAKE MENDELSSOHN To: RICHARD FOLEA

NiCds have two problems (besides price):

MEMORY—NiCds tend to "remember" how much they were used and then only operate to that level. For example, if a NiCd is discharged to its 80% level and then recharged, and if this cycle is repeated many times, then the NiCd will "remember" that 20% discharge and will not be able to go beyond it. The 20% becomes the full 100% capacity. The way around this is to "deep discharge" the NiCds every once in a while to clear their memory. I rigged up a small resistor and relay circuit to slowly discharge the NiCds and then stop when the voltage reaches a minimum. You never want to completely discharge a NiCd, because then you may get another problem: "voltage reversal."

INTERNAL SHORTS-A NiCd can also develop an internal short within the cell itself that limits its capacity. The way to clear these shorts it to charge up a large capacitor (>10,000 µF) to 50 volts and then connect the capacitor to the NiCd. There will be a large spark as the capacitor quickly discharges through the NiCd. This massive current will burn out any of the shorts. You may have to do this a few times to completely clear the NiCd. Charge the capacitor slowly with a resistor in series, but let it discharge directly through the NiCd. Be careful of the large current. It can hurt.

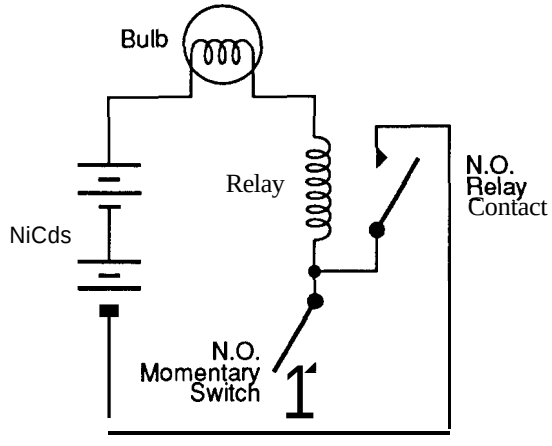
**Msg#:46170**
From: JOHN S. FETZIK To: JAKE MENDELSSOHN

What's your battery discharge circuit look like? What kind of resistor value and what voltage do you have the relay drop out at?

**Msg#:46293**
From: JAKE MENDELSSOHN To: JOHN S. FETZIK

I am NOT a battery engineer, but after many, many years of spending a small fortune on batteries, talking to many battery engineers, and reading lots of articles, here is what I have gathered:

The circuit I use for deep discharging NiCds looks like this:




Pushing the momentary switch, completes the circuit and activates the relay. The relay contacts touch thus supplying a current path when the momentary switch is released. The current from the NiCd goes through the relay coil and the bulb. When the NiCd is drained low enough that it can no longer keep the relay pulled in, the current stops. The bulb is just a visual indicator that the current is still flowing. I don't think there is any magic value on how low the voltage should be before the

relay is released. For example most 5-VDC relays, once activated, will hold their position with under 2 volts before they release.

Be sure to match the voltage required by the bulb and relay in series to the voltage of the NiCds. This supplies a slow, steady deep discharge with no possibility of completely draining the NiCd. After this deep discharge, slowly recharge the NiCds back to 100%.

**Msg#:46379**
From: DAVE EWEN To: JAKE MENDELSSOHN

That relay discharger is a very good idea. If you make it a DPDT relay you could make a cycler. I think you ought to send this schematic into a couple of the R/C magazines...maybe win something for the "best idea of the month" column.

**Msg#:46473**
From: PELLERVO KASKINEN To: JAKE MENDELSSOHN

Congratulations! You have come up with EXACTLY the same method of reviving NiCd cells as I use. The only thing that has not appeared to be quite clear on anybody's message so far is: When discharging, you better discharge each cell individually, not as a package of series-connected elements. The series connection is exactly the reason why many packages get damaged in the first place, even though they are "deep discharged." When one cell is low in charge and the whole package is loaded down, the good cells eventually force a reverse charge on the poor chap in the middle with a low or no charge.

Also, when zapping, I always do it for a single cell, even though that may not be quite as necessary as during the proper discharging.

Oh, I was going to say that the vacuum cleaners and hand tools that seem to mistreat their NiCds really bad, all have just a couple of cells in series, so the reversal is much less likely to happen: people just can not run the tool anymore once one of the two cells is empty. Therefore, no more reverse charging from several good cells into one empty cell. Or the potential available for the reverse charging, if attempted, is only that of a single cell, not several times higher.

**Msg#:46199**
From: MARC CROTEAU To: JAKE MENDELSSOHN

I have a quicker way to "ZAP" the batteries. I basically perform the same task but with something everyone has, a car! Instead of using a BIG capacitor (you might not have one handy!), I use alligator clips on my car battery and I zap the NiCd by just touching it VERY briefly.

I do this 2 or 3 times, then discharge the batteries completely and use my battery charger to fully charge the batteries.

Make sure the polarities are right (parallel) and that you don't keep the two batteries together for too long as overheating might occur and damage both batteries.


**8051 MULTITASKING EXECUTIVE**

***Develop Real Time Applications FAST!***

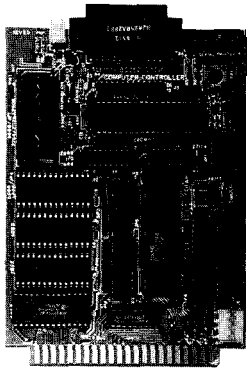
The DCE51 ™ Executive provides:

- 
- 
- 
- 
- 
- 
- 
- 
- 

**Iota Systems, Inc.**
PH: (702) 831-6302 FAX: (702) 831-4629
POB 8987
Incline Village, Nevada 89450

Reader Service #154

**BCC52**
**BASIC-52**
**COMPUTER/CONTROLLER**



The BCC52 Computer/Controller is Micromint's hottest selling stand-alone single-board microcomputer. Its cost-effective architecture needs only a power supply and terminal to become a complete development or end-use system, programmable in BASIC or machine language. The BCC52 uses Micromint's 80C52-BASIC CMOS microprocessor which contains a ROM-resident 8K- byte floating-point BASIC-52 interpreter.

The BCC52 contains sockets for up to 48K bytes of RAM/EPROM, an "intelligent" 2764/128 EPROM programmer, three parallel pods, a serial terminal port with auto baud rate selection, a serial printer port, and is bus-compatible with the full line of BCC-bus expansion boards. BASIC-52's full floating-point BASK: is fast and efficient enough for the most complicated tasks, while its cost-effective design allows it to be considered for many new areas of implementation. It can be used both for development and end-use applications.

PROCESSOR
- 80C52-BASIC, B-tit CMOS microcomputer
- jumper-selectable conversion to 80C31/80C32 functionality
- 8K bytes ROM (MI BASIC interpreter)
- 256 bytes RAM
- three 16-bit counter/timers
- 32 I/O lines
- 11 MHz system clock
- 6 interrupts

MEMORY
- expandable to 62K bytes
- five on-board sockets
- up to four 6264 (8Kx8) static RAM
- either an 8K 2764 a 16K 27128 EPROM

INPUT/OUTPUT
- console I/O RS-232 serial pal
- line printer RS-232 serial port
- three 8-bit programmable TTL-compatible parallel I/O ports using a 8255 PPI
- alternate console RS-422/RS-485

**To Order Call**
**1-800-635-3355**
Tel: (203) 871-6170
FAX: (203) 872-2204
TELEX: 643331

| | | Single Qty. | 100 Qty. |
|---|---|---|---|
| BCC52 | BASIC-52 Controller Board with 8K RAM | $189.00 | $149.00 |
| BCC52C | Lower-power all-CMOS version of the BCC52 | $199.00 | $159.00 |
| BCC52I | Full industrial temperature range | $294.00 | 5220.00 |
| BCC52CX | CMOS, Expanded BCC52 w/32K RAM | $259.00 | $159.00 |

**MICROMINT, INC.** 4 Park Street, Vernon, CT 06066

Reader Service #169

**Msg#:46266**
From: PHIL HACKETT To: RICHARD FOLEA

Most NiCds sold for competitive RC hobbyists are at least 1800 mAH. I have seen sub-Cs as high as 2300, but they are VERY expensive and are "freaks" as far as batteries are concerned. The source for these is an importer who buys them by the truckload and then sorts through them with special test equipment.

The truly competitive RC people will NOT use the same battery pack twice during the same event. The batteries will not charge as high without a 24-hour cool-down. Even a 5% loss in power is enough to make you slow.

I know this was off the subject but is something I do deal with in the hobby industry. Strange, isn't it, that "toys" are actually pushing the technological edge of these power sources.. .

*What could be more standard than RS-232? Especially when it comes to using RI-11 connectors? Here, users present some good ideas for making up cables.*

**Msg#:31074**
From: SANJAYA VATUK To: ALL USERS

Is there any kind of standard for wiring an RJ-11(6-wire) telco jack / plug for use with RS232, or does everybody do what they feel like doing? And how do you count RJ-11 pins, anyway? Inquiring minds want to know.

**Msg#:31453**
From: ROB KELLY To: SANJAYA VATUK

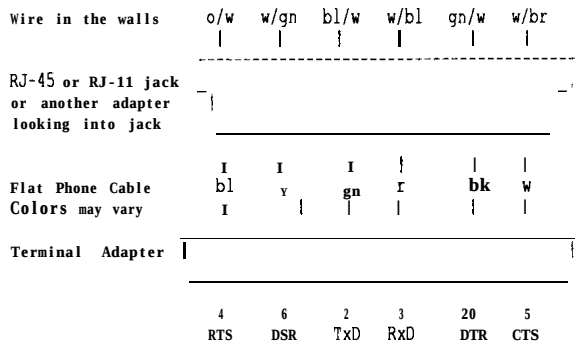
The beauty of standards is that there are so many of them...

Using the RJ-11s for RS-232 is even less standardized than RS-232 itself! Now I'll muddy the waters some more. AT&T uses RJ-45 (8-position) jacks for some of their data communication products and RJ-11 for others. They usually don't use all of the pins, though. I think there am basically two ways to go:

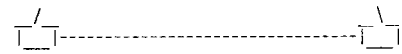
1. Wire the modular to DB-25 adapter such that flipping the flat cable to the device will provide a null modem cable. This means you wire symmetrically (i.e., TxD and RxD are the outside of the plug, RTS and CTS are on the next two pins, etc.). The advantage of this method is that all adapters are wired the same. If you need to null, you crimp the plugs with the flat cable flipped.

2. Wire such that you use a straight-through cable and wire the adapters as DTE and DCE (flip TxD and RxD, etc. in the modular to DB-25 adapter).

I am assuming that you will use flat phone cable from the wall jack to the device orbetweendevices. Also if you are coming from a wall jack that also has a phone line, you can't use the center pair. If you don't have to conform to a vendor's standard wiring plan, make your own standard! This plan uses the first method:




You the wire the flat cable like this:



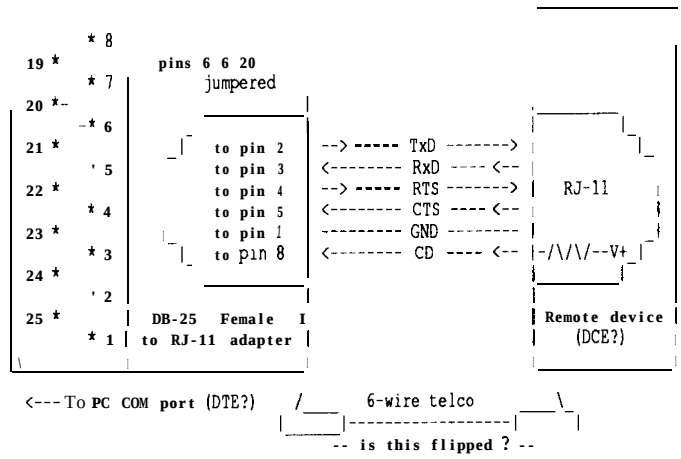
This flips the cable and performs the null. As you can probably tell I have spent a LOT of time messing with this. I hope this info helps.

**Msg#:31490**
From: SANJAYA VATUK To: ROB KELLY

Thanks! What I havedoneupuntil now is just wire the pins I need in basically numerical order as they are numbered on a DB-25. I was using flipped cable because I hadn't planned on making my own, so my female adapters were opposite the male. I think. I'll get back to you when I figure out what I did versus what you suggest.

**Msg#:31608**
From: SANJAYA VATUK To: ROB KELLY

All I can say is this had better come out formatted the way I drew it on my machine! It only tookan hour. :-) Here's how I wired my setup:




Hope this makes sense.

What is CD? And do I need it for my PC corn port to work properly? If not, it seems I could connect DSR straight through

instead (pin 6). I don't need DTR in the remote device. Thanks for your help!

**Msg#:31752**
From: ROB KELLY To: SANJAYA VATUK

Yes, that cable flips. Hold both plugs side by side, the wire colors should be in the same order. I left off one detail on my diagram: ground! You would have two ground pins to keep things symmetrical. Sorry about the error.

**Msg#:31787**
From: SANJAYA VATUK To: ROB KELLY

Thanks! I do have one ground. I ended up deleting DCD and adding DTR, or whatever pin 6 is on a DB-25.

**Msg#:31636**
From: KEN DAVIDSON To: SANJAYA VATUK

Very nice. Do you give lessons in ASCII graphics? As you've found, if you leave at least one space at the beginning of each line, the system won't try to reformat the message when it displays it.

CD (or often DCD) stands for Carrier Detect (or Data Carrier Detect). Your PC shouldn't need it in order to talk to the device.

**Msg#:31661**
From: SANJAYA VATUK To: KEN DAVIDSON

Thanks! I was going to try using box characters, but I doubt it would work. I'll try rewiring my gizmos without CD and see if it works.

**Msg#:31708**
From: KEN MUNKHOLM To: SANJAYA VATUK

Check the back page of the latest CCINK. GTEK has a line of RS-232 boards that have RJ-11 connections. You might want to call them for some information regarding pinout standards. That way you don't lock yourself out of some options for hardware down the road.

**Msg#:31528**
From: ED NISLEY To: KEN DAVIDSON

Ken, if you don't print this in ConnecTime, I want to hear the excuse. Don't tell me the message went to the wrong person, I know you read all of 'em anyway.

*The Circuit Cellar BBS runs on a 10-MHz Micromint OEM-286 IBM PC/AT-compatible computer using the multiline version of The Bread Board System (TBBS 2.1M) and currently has four modems connected. We invite you fo call and exchange ideas with other Circuit Cellar readers. It is available 24 hours a day and can be reached at (203) 871-1988. Set your modem for 8 data bits, 2 stop bit, and either 300, 2200, or 2400 bps. For those callers with USRobotics Courier HST 9600-bps modems, call (203) 871-0549. Please note this line is not compatible with V.32 modems at 9600 bps.*

## IRS

*431* Very Useful
432 Moderately Useful
433 Not Useful


**SOFTWARE AVAILABLE** on DISK

Software on Disk
Software for the articles in this issue of Circuit Cellar INK may be downloaded free of charge from the Circuit Cellar BBS. For those unable to download files, they are also available on one 360K, 5.25" IBM PC-format disk for only $12.

Circuit Cellar BBS on Disk
In the past, messages from the Circuit Cellar BBS have been available on disk. This service has been discontinued for new messages, but older messages are still available. Please refer to past issues of Circuit Cellar INK for more information.

To order Software on Disk, send check or money order to:

Circuit Cellar INK
Software on Disk
P.O. Box 772
Vernon, CT 06066

or use your Mastercard or Visa and call (203) 8752199. Be sure to specify the issue number of each disk you order. Please add $3 for shipping outside the U.S.



68HC11

DATA LOGGER

New PC driver converts WICE68HC11 to a real time data logger. Saves target data in a file while running user program.




- PC based user friendly ICE with data logging capability.
- Real time and full speed up to 14MHz clock rates.
- Single chip and expanded modes.
- Data watchwindows for memory, registers & stack.
- On-board 64K emulation RAM maps in 4K blocks.
- 64K real time hardware breakpoints.
- Breaks on address, address range, and memory RD/WR.
- Full symbolic debugging.
- Supports all A,E and D parts.
- 115.2K bps RS-232C link.
- 30 day money back guarantee.

WICE 68HCll emulator $795.00
52 PLCC to 48 DIP adapter $55.00
Call: (708) 894-1440

Wytec
Suite 140
185C E. Lake Street
Bloomingdale, IL 60108

Z8 — WICE Z8 emulator $995
86C08 adapter w/analog comparators $55

2500AD — We sell all 2500AD software products. Call us for l

Reader Service #210

# STEVE'S OWN INK

Steve Ciarcia

# Consultant's Dilemma

The Circuit Cellar BBS is more than a place for readers to exchange technical information. Frequently it has become a place where consultants and business entrepreneurs discuss basic business techniques and ethics.

Recently, one of the consultants on the BBS had a potential customer who wanted to evaluate a design but would only entertain buying the product after it was designed. Part of the deal also included building a prototype for evaluation. Because he didn't trust them, however, he was asking our advice on protecting his design. Was it acceptable to sand off the numbers on ICs? Should he pot both the prototype and the finished units so that they wouldn't steal his design?

Much to my amazement, messages poured in describing the best ways to remove IC nomenclature, complete with discussions on the particular merits of using special solvents instead of abrasives.

Often, the BBSers are so quick to answer a question or offer advice that they don't look at the whole picture. I think everybody completely missed the boat. In my opinion, the first rule of business (consulting or otherwise) is that if you don't trust the customer, you shouldn't even be there. If the design is truly customized for a particular application, albeit inexpensively reproduced, you have invested your time and money already. It's called risk.

Whether or not the customer buys it depends upon their need and willingness to have you as the sole supplier. If distrust is at the level where a "rip-off" is feared, presenting disguised designs only fuels unethical behavior. The customer's perception will be that your solution is so inexpensive (even if you bargain price the finished unit) that they will certainly entertain an in-house design at that point. And, because all engineers share similar training, their solution might end up being very similar. Legally this may not be a "rip-off."

I am not advocating that consultants should not do designs on faith or risk their efforts in designing custom products. Instead, I am advocating speculation when it is strengthened by a trusting relationship with a customer. You should have an agreement in writing up front that acknowledges your efforts toward a product design and that should the company market said device or any part thereof that you are due proper consideration. I'm not telling you how to write a contract, just that in lieu of an ongoing relationship that has already defined these parameters in detail, no reasonable company should refuse putting this agreement in writing.

My experience has been that reasonable-sized businesses cannot deal with black-box purchased products unless complete warranty responsibility is in the hands of the supplier. A one-man shop will have difficulty doing warranty equipment swaps in Saudi Arabia and keeping production rolling at the same time.

Often companies can't invest in new designs as often as they like because the process is too cumbersome or political. Just writing a specification can often take six months. However, "showing them a product" can create double jeopardy. What they may not have been able to specify adequately before will now be much clearer in their minds. The clock starts ticking then toward the ultimate release of that product whether it comes from you or them. Unfortunately, if you don't have adequate resources to produce the product or warranty as required, you will be on the outsideagain. This is not being ripped off. It's called learning free enterprise the hard way.

Except for being wary when you find yourself using words like "steal" or "protect," you should not avoid taking risk and investing design time for reputable customers. First of all, while you may have graduated well up in your class, there are always smarter people. Count on the fact that a bunch of them work for the customer. Anything you design, they can do too. Your best move is openness. The engineers evaluating your design are employees, not owners. Holding back information only makes their job harder. The more information you provide and document, in fact, the harder it will be for them to substantiate claims of an independent design arriving at the same conclusion.

Finally, unless you have the manufacturing resources to meet the cost, delivery, and warranty requirements of the company, don't even try. Don't give them a prototype that triggers their "designcycle clock" without a plan to benefit from it. Don't think sole ownership. Instead, offer to license or sell the complete design on a one-time fee or royalty basis.

It's a hard lesson to realize that our professional contribution might be quite small when considered in the grand scheme of things. Every design can't be the next hoola hoop any more than every lottery ticket can be a winner. Consulting is a combination of sense and substance. The truly successful consultant is the one who understands how to look at business through the eyes of his customer.

