

CIRCUIT CELLAR

I N K[®]

THE COMPUTER APPLICATIONS JOURNAL

April 1994 — Issue #45

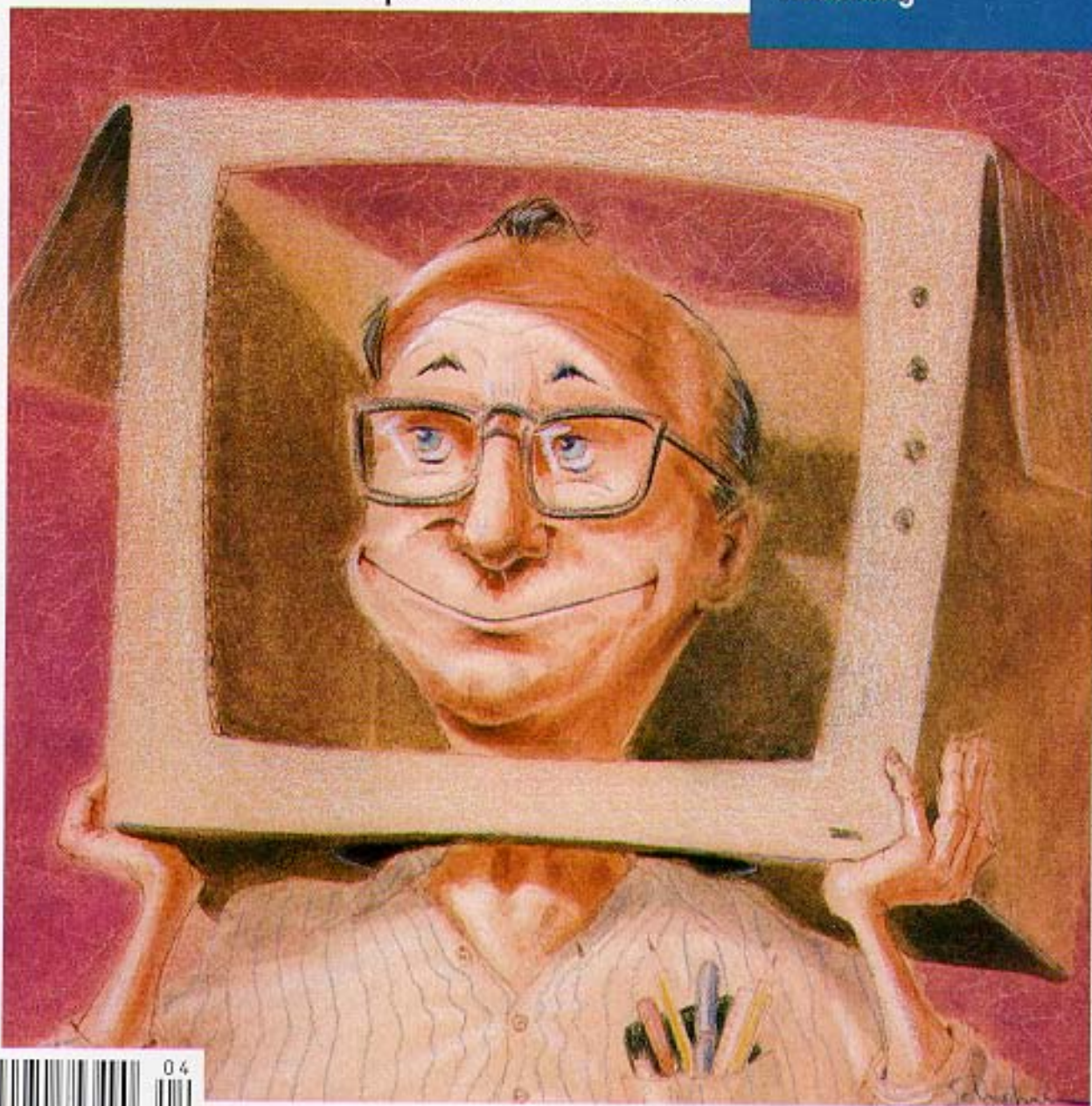
GRAPHICS & VIDEO

Ciarcia's New
Audio/Video Mux

Explore the Vertical
Blanking Interval

Who Owns the Work You
Do at Home?

Solid-state Audio
Recording



\$3.95 U.S.
\$4.95 Canada

Video Galore



With video sources—over-the-air TV, cable TV, VCRs, videodiscs, cameras, multimedia PCs—proliferating today's households and businesses like never before, it's becoming increasingly difficult to manage all these sources. To make matters worse, what happens when you want to direct multiple sources to multiple destinations, all with a single switchbox and no recabling? Enter Steve's newest project: the **AVMux**.

Steve first introduced an AVMux back in 1986 and it was an instant hit. Now he revisits the old concept with some new ideas, chips, and remote control schemes. We start this month with the first part of his project: the multiplexer itself.

Along with the latest wiz-bang 500-channel cable systems comes the question, "But is there anything worthwhile watching yet?" (See "Steve's Own INK" on the last page for more on this.) If you read between the lines, though, you can find worthwhile information hidden inside even off-air TV signals. Contained in the vertical blanking interval, our next project is designed to ferret out such goodies as closed captions, teletext publications, and time of day. This design contest winner should keep you busy for hours.

Our last feature should be of profound interest to anyone who does work at home, has work-related hobbies, or does work for hire (I think I just described the vast majority of our readers). Who owns the copyright on that work you do in your home or on your own time? It's a sticky issue in some cases, so we found two lawyers who specialize in intellectual property to look at recent case law and try to smooth it over for us.

In our columns, Ed continues with his large LCD panel interface by adding some real software support. Jeff revisits computerized speech by putting together a small, recordable audio output board that can be used for speech and sound effects alike. Tom looks at a new processor using a super Harvard architecture (SHARC) that sports no fewer than four address/data buses internally. Finally, John adds a reset circuit, battery monitor, and LCD display to his embedded controller system.

On a final note, I want to mention that this is the last installment of Russ Reiss's "Patent Talk." Russ has covered many diverse topics and has done a fine job of pointing out just how much help our patent system needs if it's to keep up with the explosive growth of today's technology. In his last column, Russ looks at security-related devices, from the standpoint of both data security and system security.

THE COMPUTER APPLICATIONS JOURNAL

FOUNDER/EDITORIAL DIRECTOR
Steve Ciarcia

PUBLISHER
Daniel Rodríguez

EDITOR-IN-CHIEF
Ken Davidson

PUBLISHER'S ASSISTANT
Sue Hodge

TECHNICAL EDITOR
Michael Swartzendruber

CIRCULATION COORDINATOR
Rose Mansella

ASSOCIATE EDITOR
Rob Rojas

CIRCULATION ASSISTANT
Barbara Maleski

ENGINEERING STAFF
Jeff Bachiochi & Ed Nisley

CIRCULATION CONSULTANT
Gregory Spitzfaden

WEST COAST EDITOR
Tom Cantrell

BUSINESS MANAGER
Jeannette Walters

CONTRIBUTING EDITORS
John Dybowski & Russ Reiss

ADVERTISING COORDINATOR
Dan Gorsky

NEW PRODUCTS EDITOR
Harv Weiner

CIRCUIT CELLAR INK, THE COMPUTER APPLICATIONS JOURNAL (ISSN 0896-8985) is published monthly 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 (12 issues) subscription rate U.S.A. and possessions \$21.95, Canada/Mexico \$31.95, all other countries \$49.95. All subscription orders payable in US funds only, via international postal money order or check drawn on US bank. Direct subscription orders and subscription related questions to The Computer Applications Journal Subscriptions, P O Box 7694, Riverton, NJ 08077 or call (609) 786-0409. POSTMASTER: Please send address changes to The Computer Applications Journal, Circulation Dept., P.O. Box 7694, Riverton, NJ 06077

ART DIRECTOR
Lisa Ferry

GRAPHIC ARTIST
Joseph Quinlan

CONTRIBUTORS:
Jon Elson
Tim McDonough
Frank Kuechmann
Pellervo Kaskinen

Cover Illustration by Bob Schuchman
PRINTED IN THE UNITED STATES

HAJAR ASSOCIATES NATIONAL ADVERTISING REPRESENTATIVES

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

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

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

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

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

Circuit Cellar BBS—24 Hrs. 300/1200/2400/9600/14.4k bps, 8 bits, no parity, 1 stop bit, (203) 871-1988; 2400/1600 bps Courier HST, (203) 871-0549

All programs and schematics in *Circuit Cellar INK* have been carefully reviewed to ensure their performance 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 liability of any kind for errors in these programs or schematics or for the consequences of any such errors. Furthermore, because of possible variation in the quality and condition of materials and workmanship of reader-assembled projects, *Circuit Cellar INK* disclaims any responsibility 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 © 1994 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.

1 4 Control Your Audio/Video Connections
with the **AVMux**
by Steve Ciarcia

2 4 Exploring the Vertical Blanking Interval/A Tool for
Gathering Data from your TV
by Mike Barnes

4 0 Employer Ownership of Employee and Consultant
Work Product
by Mary Luria @ Laura Butzel

4 4  Firmware Furnace
Bringing the '386SX Project's
Graphic LCD Panel to Life
Ed Nisley

5 4 From the Bench
Build the Message Board/An Audio
Record/Playback Unit
Jeff Bachiochi

6 2 Silicon Update
When the SHARC Bites
Tom Cantrill

6 8 Embedded Techniques
Who's in Control?
John Dybowski

INSIDE ISSUE 45

2 Editor's INK
Ken Davidson
Video Galore

6 Reader's INK
Letters to the Editor.

8 New Product News
edited by Harv Weiner

77 Patent Talk
Russ Reiss

ConnectTime **84**
Excerpts from
the Circuit Cellar BBS
conducted by
Ken Davidson

Steve's Own INK **96**
Steve Ciarcia
In the Citadel

Advertiser's Index **81**

READER'S INK

Let Me Interrupt You for a Moment

I have just finished reading issue #43, and I have strong feelings regarding Do-While Jones' article ("Interrupt-free Design"). Although the underlying premise (that design decisions must be made thoughtfully, not just because things have always been done that way) is sound, I feel that Mr. Jones has greatly overstated his case, perhaps for effect. There are several valid points made in the article, but I feel that they are all but obscured by the "over the top" rhetoric and use of inappropriate analogies.

Mr. Jones extracts interrupt usage from the rest of the firmware design context, and then expounds on its inappropriateness. The creation of a design is a holistic process, wherein all the issues have to be weighed together and the "best" design is obtained by applying tradeoffs. Whether or not to use an interrupt for a particular peripheral is but one of these many tradeoffs.

His A/D conversion example makes multitudinous (unstated) assumptions about the application at hand and the design of the rest of the firmware. Perhaps these assumptions are valid for the case that he had in mind, but this is hardly justification for sweeping generalizations against interrupts. For example:

"An A/D converter satisfies neither of the two criteria for correct use of an interrupt. Interrupts should be used when (1) you can't predict when data will be available and (2) data must be taken from the peripheral promptly or it will be lost. Dedicated controllers always know when data from an A/D converter will be available. It comes at a periodic rate synchronized to a clock. It should be no surprise that the data is available. Furthermore, it is the program that must have the data promptly, not the peripheral that must have the data taken from it promptly."

If it were true that you can always predict when data is available from an A/D converter, why is it that they are equipped with a "conversion done" output? For Mr. Jones to assert that it is not the ADC that requires the data be taken promptly, it is clear he has never had to run an ADC in continuous conversion mode, where the output must be captured quickly or it will be replaced by the next sample.

Based on his background, I am sure Mr. Jones' examples of "justified use of interrupts" don't accurately reflect his understanding of design issues. Consider:

". . . For example, the radiation-leak detector in a nuclear power plant should be connected to an interrupt...."

Here he confuses system-level design issues (i.e., the response time) with controller-level issues (i.e., interrupt

input or not.) For all we know, the emergency detection system could be a dedicated controller responsible for monitoring 100 digital inputs, displaying their state on 100 indicators, and activating a horn as an alarm. In this case, the simplest implementation is a continuous loop that polls the inputs, reflects their state in the indicators, and enables the horn output based on some simple logic conditions. No need for interrupts here!

In contrast, consider a hardware/firmware design I recently completed. Among other peripherals is a 12-bit dual-slope ADC taking 16.6 ms per conversion. The ADC is not directly connected to an interrupt input, but is polled at an interval determined by a timer interrupt. While it would be possible to perform the same work in the "background" processing, I have selected the method I did because it allows me to know precisely the rate at which the inputs are sampled. This wouldn't be easy to do if the polling were done in a background process.

In my experience, firmware designers are too *unwilling* to use interrupts, because (and I agree with Mr. Jones on this point) they are more difficult to understand. Unfortunately, their avoidance often results in firmware that, as a whole, is more complex, harder to debug, and harder still to maintain.

The points to remember, with my annotations:

@ "Knowing how to recognize situations when it is wise to use interrupts or to avoid them like the plague will lead to. . . [a] design that may be cheaper, simpler, and more reliable." [I disagree with the assertion that the interrupt-free design will always be the one that is cheaper, simpler, etc.]

• "It is always faster to poll a peripheral's status port to make sure it is ready and then transfer data to (or from) the peripheral's data port than it is to respond to an interrupt generated by the peripheral's status port." [But this is a very local consideration, that could easily be overridden by higher-level issues.]

* "Interrupt-driven designs tend to be chaotic and nondeterministic. . . you have to be very careful to make sure there are no race conditions or sequences of interrupts that can cause trouble. . . Interrupt-free designs are much more predictable. . . The logic is simpler because the program controls the order in which things happen." [There is no doubt that interrupts introduce difficulty, and are not necessary in every case.]

• ". . . interrupts should be used when they make the design simpler. . . If you were using interrupts appropriately, they would solve your problems, not cause them."

All in all, the article would have been much better if it presented a balanced view and tried to give realistic case histories where interrupts were and were not

READER'S INK

appropriate. As it stands, it comes across as a tirade against interrupts, and an unconvincing one at that.

Martin Vuille
Kemptville, Ontario
martin.vuille@synapse.org

And While We're on the Subject...

I thoroughly enjoyed Do-While Jones' article in the February 1994 issue. Never have truer words been written about the abuse of interrupts.

When Mr. Jones was a speaker at the Embedded Systems Conference, I remember finding his short biography very amusing. It described his progression from analog designer to lecturer. The next time you talk to him, ask Mr. Jones if he has any other loop constructs in his family.

J. Conrad Hubert
Deus Ex Machina
St. Paul, Minn.

Contacting Circuit Cellar

We at the *Computer Applications Journal* encourage communication between our readers and our staff, so have made every effort to make contacting us easy. We prefer electronic communications, but feel free to use any of the following:

Mail: Letters to the Editor may be sent to: Editor, The Computer Applications Journal, 4 Park St., Vernon, CT 06066.

Phone: Direct all subscription inquiries to (609) 786-0409.

Contact our editorial offices at (203) 875-2199.

Fax: All faxes may be sent to (203) 872-2204.

BBS: All of our editors and regular authors frequent the Circuit Cellar BBS and are available to answer questions. Call (203) 871-1988 with your modem (300-14.4k bps, 8N1).

Internet: Electronic mail may also be sent to our editors and regular authors via the Internet. To determine a particular person's Internet address, use their name as it appears in the masthead or by-line, insert a period between their first and last names, and append "@circellar.com" to the end. For example, to send Internet Email to Jeff Bachiochi, address it to jeff.bachiochi@circellar.com. For more information, send Email to info@circellar.com.

TURBO-1 28 SINGLE BOARD BASIC DEVELOPMENT SYSTEM

The Photronics Research T-128 **SBC** features Dallas Semiconductor's new **8051-compatible DS80C320 high-speed microcontroller**. With its 2X clock speed (25 MHz) and 3X cycle efficiency an instruction can execute in 180ns: an 8051 equivalent speed of **62.5MHz!!!**

The T-128 board **NEEDS NO EPROM EMULATOR, PROGRAM MER OR ERASER**. With our unique on-board memory management circuitry, a 70ns 128Kx8 SRAM is configured into any nonvolatile/write protectable combination of CODE, DATA, or **OVERLAID** mapping.

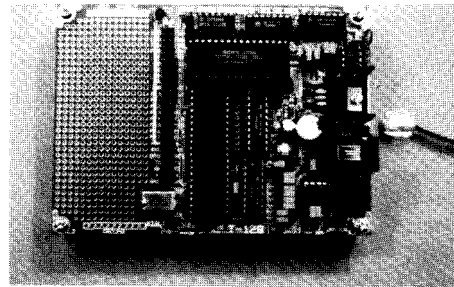
- All RAM is programmed on-board
- Dual Cell, Embedded Lithium Backup Technology
- Crashproof

NEW ON CHIP FEATURES INCLUDE:

- Programmable Watchdog Timer
- Early Warning power monitor
- Two full-duplex serial ports
- Duel data pointers
- 13 interrupt sources (6 ext., 7 int.)

ACCELERATED BASIC-52

- Modified Basic-52 now fast enough for new applications
- Stack BASIC programs in RAM and autorun
- 8 bit ports and device-ready I/O
- Only 5" x 3-3/4" includes 500-hole proto area
- One 4-wire telephone cable connects console/power
- The powerful combination of an accelerated Basic/Assembly Interface easily approaches 16 bit capacity
- 8-MIP "NITRO" upgrade available soon



With assembler, editor, terminal emulator, utilities
1 year warranty • Free technical support
Manuals and power adapter included • Ready to Run!
INTRODUCTORY PRICE \$239

Photronics Research, Inc. • 1 OS Camille St. • Amite, IA 70422 • 15041 748-9911

NEW PRODUCT NEWS

Edited by Harv Weiner

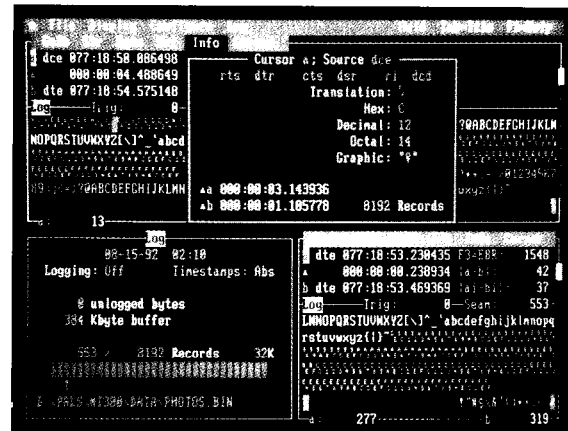
SERIAL LINE MONITOR

Paladin Software has announced a new version of **MicroTAP** (formerly DataScope), a software-based asynchronous communications debugging, data capture, and analysis tool. MicroTAP allows a user to apply capture, display, and search tools to ordinarily invisible serial transmissions and provides a cost-effective alternative to expensive hardware-based line monitors.

MicroTAP includes context-sensitive Hypertext with direct links to program setup fields, HyperAccess, user-alterable multitasking window displays, oscilloscope-like signal event tracing, an integrated font map editor, and PostScript file exportation (EPS, PRN, TXT, or binary files of logged data for printing or importing into reports). All data and signal events are timestamped to the microsecond. A log capacity of 64 megabytes is provided and a RAM buffer of up to 384K bytes allows logging to disk at any time.

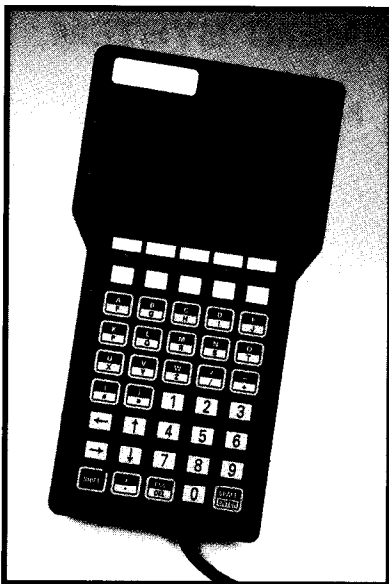
MicroTAP supports all COM ports and UART formats. Any port or interrupt level may be used and two ports can share the same interrupt line. All possible baud rates supported by the hardware are available and automatic nearest true baud rate adjustment is used. All combinations of word length, parity, stop bits, and output control lines are supported.

MicroTAP offers a convenient carrying case complete with storage space for the manual, cable, and connectors. A tutorial answers common questions and helps users get started with a minimum amount of reading. MicroTAP sells for \$349.



Paladin Software, Inc. • 3945 Kenosha Ave. • San Diego, CA 92117 • (619) 490-0368 • Fax: (619) 490-0177

#500



MULTIDROP INDUSTRIAL TERMINAL

The QTERM-IV Industrial Terminal is a low-cost, rugged, operator-interface terminal capable of multidrop operation (RS-485 or RS-422). The terminal features either full-duplex or multidrop operation with up to 32 terminals per host port, a 4-line by 20-character supertwist LCD display, 24- or 40-key tactile keypads, and user-programmable LEDs.

The QTERM-IV can be ordered with a choice of RS-232, RS-422, RS-423, RS-485, or multidrop- interfaces. Every key can be individually programmed to send a character or string (or to control QTERM-IV) when pressed and/or when released. User-programmable LEDs provide easy and flexible status indications. Forty-eight programmable macro strings and an autoexecute string allow for fast and easy configuration of menu systems. Up to 20K bytes of user data can be stored in the unit.

The QTERM-IV can be configured for an application by power-on setup or by downloading a data file. After configuration, the host can control it via software commands. Available commands include cursor movement, mode control, query, hardware control, macros, and user data read and write commands. Custom characters can also be defined.

The QTERM-IV measures 7.7" high, 4.2" wide, and 1.3" thick. It uses less than 20 mA of current from a 5-volt supply. It can be powered through the host connector or can be ordered with a battery option. Batteries can supply power for over 100 hours of continuous use.

The QTERM-IV sells for \$295. Custom OEM labeling, including a company name, user LED labels, and one row of function keys, are included free with any quantity purchase. Full-custom keypad graphics are also available for a modest additional price.

QSI Corporation • 2212 South West Temple #46 • Salt Lake City, UT 84115 • (801) 466-8770 • Fax: (801) 466-8792

#501

NEW PRODUCT NEWS

SCSI IN PCMCIA PACKAGE

True "Plug and Play" SCSI in a PCMCIA Type II format has been announced by Future Domain. The **SCSI2GO** is a credit-card-sized, 16-bit SCSI controller. It is an easy way to add SCSI storage devices and I/O peripherals to notebook PCs. It can also be used with any PC containing a standard PCMCIA Type II card slot, including desktop PCs, PC workstations, and PC servers.

The controller is compatible with Card and Socket Services software. It is capable of fast SCSI-2 data transfer rates up to 10 MB/s. SCSI2GO features Future Domain's high-performance 18C30 single-chip SCSI-2 controller with a 2K internal FIFO buffer. The three-foot interface cable is a high-quality SCSI cable with a locking PCMCIA card connector on one end and a 50-pin SCSI-2 connector on the other.

In keeping with the power-saving philosophy of notebook PC design, SCSI2GO has a powered-down Sleep Mode. While operating, the controller has a very low power requirement of only 210 mW.

SCSI2GO is fully supported by PowerSCSI!, the universal application interface under DOS, NetWare, and Windows. Embedded support is provided in Windows NT and OS/2.

SCSI2GO is available in SCSI VALUEPAK and CorelSCSI Kit configurations. The SCSI VALUEPAK Kit sells for \$329 and includes the SCSI2GO card, interface cable, PowerSCSI! software, and additional utility software. The CorelSCSI Kit sells for \$389 and contains CorelSCSI software.



Future Domain Corp. • 2801 McGaw Ave. • Irvine, CA 92714 • (714) 253-0400 • Fax: (714) 253-0913

#502

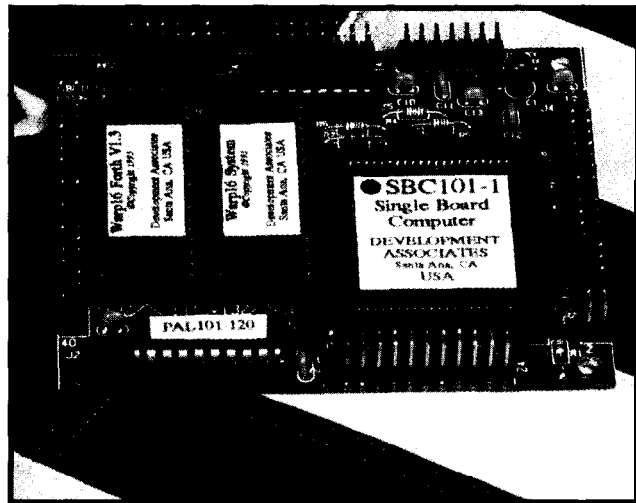
SINGLE-BOARD COMPUTER

Development Associates has announced a series of small, cost-effective 16-bit single-board computers. The Warp System is intended for a wide range of embedded system applications. These SBCs are based on the high-integration CMOS V25 microcomputer and are available with clock rates of 10–20 MHz.

The SBCs are fully self-contained and feature all CMOS components for low power consumption. They are available in ROM ranges from 16K bytes to 512K bytes and RAM capacities from 32K bytes to 512K bytes. The CPU addresses are fully decoded, making all unused on-board addresses available for external devices. System expansion is simplified since all CPU signals have been brought out to connector pins. The SBCs are 3" x 3.5" and are powered by a single 5-V power source.

The SBCs feature reset and power-on signal conditioning, watchdog and low-voltage detection functions, two serial ports usable to 1.2 Mbps, and eight analog comparator inputs with programmable thresholds. Additional features include 24 parallel I/O lines, two 16-bit timers, timebase counter, and a programmable wait state generator.

The SBCs are software compatible with the 80186/88 family but include enhanced instructions. They are compatible with a large number of software tools including Development Associates' Warp16 Forth and WarpP16 Future86. Both of these development environments are totally complete, providing for compiling, linking, debugging, hex file generation, and other development tasks. The Warp System SBC101 series starts at \$119. A complete line of accessories is also available.



Development Associates • 1520 South Lyon St. • Santa Ana, CA 92705 • (714) 835-9512

#503

NEW PRODUCT NEWS

CHEMICAL CIRCUIT PROTECTOR

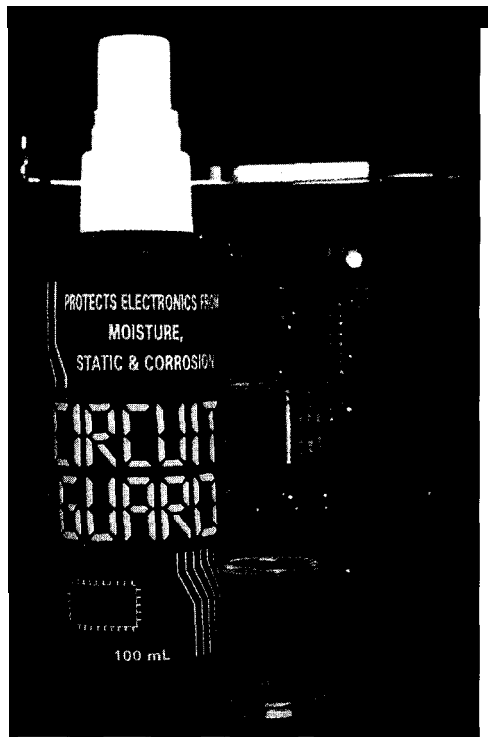
A newly formulated product that protects computers and electronics from moisture, static, and corrosion has been announced by Circuit Guard International. Circuit **Guard** was originally developed for the aerospace industry to protect sensitive microcircuits from exposure to harsh environments.

Circuit Guard forms an atmospheric barrier around electronic components. This barrier actively encapsulates and chemically neutralizes all ion-transferring electrolytes (such as moisture, pollutants, salt, and dust) which cause corrosion leading to circuit failure. Circuit Guard will also safely discharge harmful static by neutralizing charged particles.

The spray-on liquid is compatible with the plastics, epoxies, and various metals used in electronic systems. Circuit Guard has a dielectric strength in excess of 38,000 volts and does not alter resistive, capacitive, or inductive circuits. It is emulsifiable in water, is nontoxic and nonirritating to skin and eyes. Circuit Guard is available in 100-mL nonaerosol spray bottles for \$19.95.

Circuit Guard International
 3650 Silver-side Rd. • S. Wilmington, DE 19810
 (905) 509-8752 • Fax: (905) 509-4491
 Internet: circuitguard@delphi.com

#504



ATTEND THE SECOND ANNUAL



JUNE 9-10, 1994
 SANTA CLARA
 CONVENTION CENTER
 SANTA CLARA
 CALIFORNIA

Hardware and Software Solutions for Embedded Computer Applications

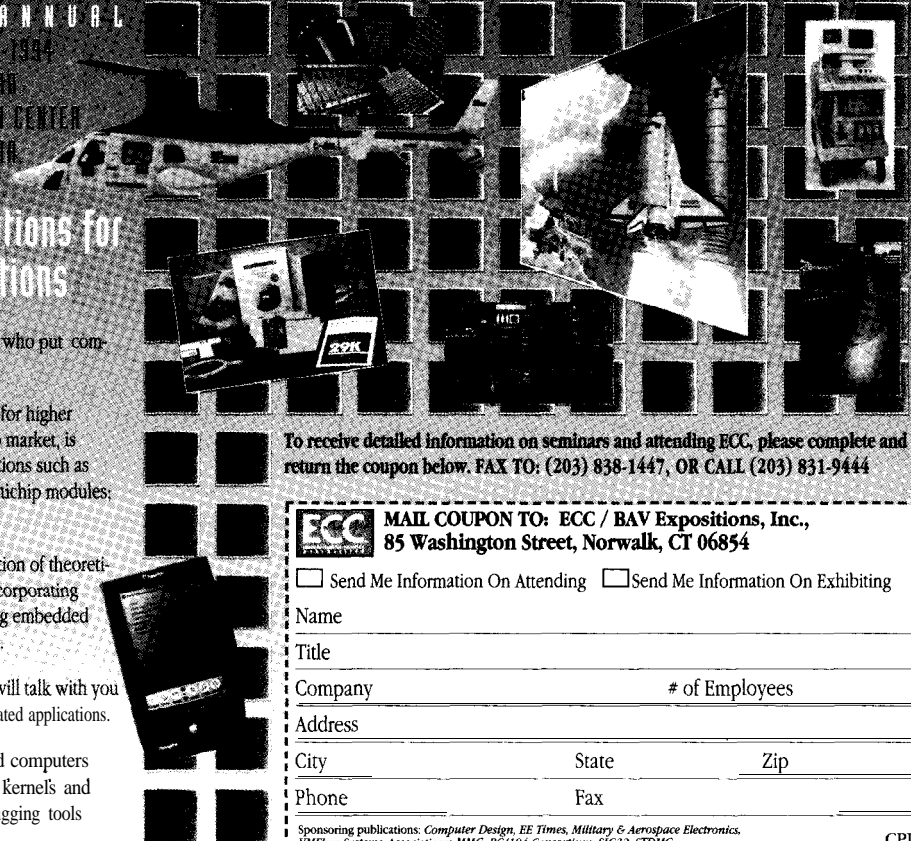
Who Should Attend: Hardware and software engineers who put computing power to work in embedded applications.

Technical Program "32 bits or better" The demand for higher performance, smaller size, lower cost and faster time to market, is forcing the move to more powerful and advanced solutions such as 32- or 64-bit RISC or DSP for processing; ASICs and multichip modules; high-performance backplane buses and more...

You'll find two and a half days packed with a combination of theoretical, practical, and application orientated sessions on incorporating high-performance embedded computers and developing embedded realtime software for complete systems and subsystems.

Exhibit Hall: A comprehensive mix of leading vendors will talk with you about solutions to your specific embedded and dedicated applications.

See The Latest In: microprocessors single-board computers peripheral boards operating systems realtime kernels and compilers development systems CASE and debugging tools embedded PCs and more...



To receive detailed information on seminars and attending ECC, please complete and return the coupon below. FAX TO: (203) 838-1447, OR CALL (203) 831-9444



MAIL COUPON TO: ECC / BAV Expositions, Inc., 85 Washington Street, Norwalk, CT 06854

Send Me Information On Attending Send Me Information On Exhibiting

Name _____

Title _____

Company _____ # of Employees _____

Address _____

City _____ State _____ Zip _____

Phone _____ Fax _____

Sponsoring publications: Computer Design, EE Times, Military & Aerospace Electronics, VMEbus Systems. Associations: MMG, PC/101 Consortium, SIG32, STDMG.

CPJ

NEW PRODUCT NEWS

REAL-TIME PC CONTROL COPROCESSOR

Modular Micro Controls has introduced the PC-452, an 8051-family-compatible single-board controller configured as a slave coprocessor for the PC/AT ISA bus. Typical applications include network master controller, intelligent serial port controller, I/O controller, and peripheral controller.

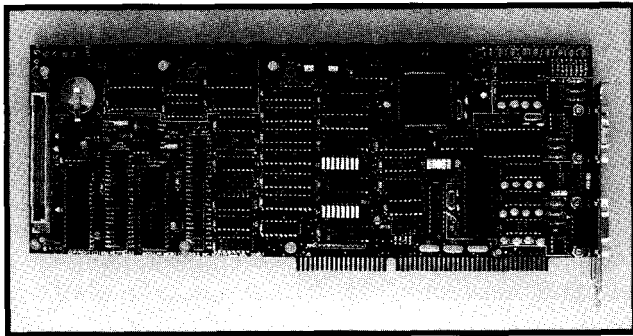
The board has been developed around Intel's 80C452 Universal Peripheral Interface microcontroller and interfaces to the PC bus with a 128-byte bidirectional FIFO. The board features two 8-bit, high-current, quasi-bidirectional I/O ports; two CMOS MPU I/O

ports; two 8-bit user DIP switches; and an 8051-compatible Y-bit UART with either RS-232C or RS-485 drivers. The PC-452 provides a standard 8051 local bus with two JEDEC-compatible, 32-pin memory sites (Code and Data). The standard configuration includes 64K of EPROM and 32K of battery-backed SRAM.

To aid in debugging software, a status LED is provided for each bit of MPU Port 1. An MC-bus expansion connector and an external power supply input connector are also provided. The expansion connector is compatible with Modular Micro Controls' line of peripheral modules.

The PC-452 can be ordered with a second UART, RS-485 ARCNET interface, and Flash memory programmer for the 8051 Code memory site. It comes complete with a debugger, editor, assembler, BASIC and Forth, serial terminal, and FIFO terminal emulators for both Windows and DOS. The PC-452 board with standard features and development tools sells for \$499.

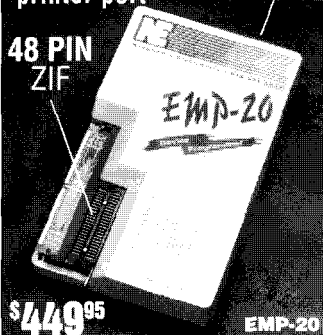
Modular Micro Controls, Inc. • 411 S. Water St.
Northfield, MN 55057 • (800) 832-7731 • Fax: (507) 645-4342
#505



Device Programmers

Connects to PC parallel printer port

48 PIN ZIF



\$449⁹⁵

EMP-20

- Easy to use software, on-line help, full screen editor
- Made in USA
- 1 & 2 Year Warranty
- Technical Support by phone
- 30 day Money Back Guarantee
- FREE software upgrades available via BBS
- Demo SW via BBS (EM20DEMO.EXE)(PB10DEMO.EXE)
- E(e)proms 2716 8 megabit, 16 bit 27210-27240, 27C400 & 27C800,
- Flash 28F256-28F020,(29C256-29C010 (EMP-20 only))
- Micros 8741A, 42A, 42AH, 48, 49, 48H, 49H, 55, 87C51, 87C51FX, 87C751.752
- GAL, PLO from NS, Lattice AMD-16V8, 20V8, 22V10 (FMP-20 only)

SA-20 8 Gang Eprom

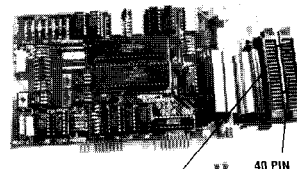
20 X 4 Line LCD Display



\$750⁰⁰

20 Key Keypad

PB-10 Internal Card for PC



\$139⁹⁵

2 ft. Cable

40 PIN ZIF

The only 8051/52 BASIC compiler that is 100% BASIC 52 Compatible and has full floating point, integer, byte & bit variables.

- Memory mapped variables
- In-line assembly language option
- Compile time switch to select 8051/8031 or 8052/8032 CPUs
- Compatible with any RAM or ROM memory mapping
- Runs up to 50 times faster than the MCS BASIC-52 interpreter.
- Includes Binary Technology's SXA51 cross-assembler & hex file manip.util.
- Extensive documentation
- Tutorial included
- Runs on IBM-PC/XT or compatible
- Compatible with all 8051 variants
- BXC51 \$ 295.

508-369-9556

FAX 508-369-9549



Binary Technology, Inc.

P.O. Box 541 • Carlisle, MA 01741



FOR MORE INFORMATION CALL

NEEDHAM'S ELECTRONICS, INC.

4539 Orange Grove Ave.
Sacramento, CA 95841
(Monday-Friday, 8 am-5 pm PST)



C.O.D.

(916) 924-8037
BBS (916) 972-8042
FAX (916) 972-9960

NEW PRODUCT NEWS

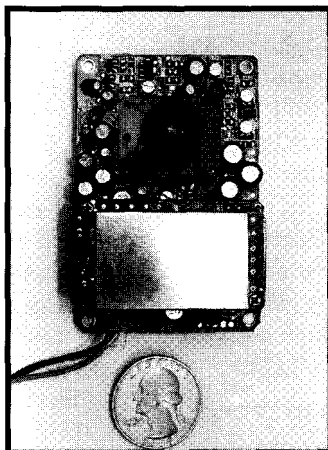
MINIATURE VIDEO CAMERA

The Optical Systems Division of Marshall Electronics has introduced a micro-size, high-tech, black-and-white CCD camera module. The Model V1206DC can be used in such applications as teleconferencing, multimedia, machine vision, inspection, and monitoring.

The Model V1206DC can also be used as a microscope or magnifying device in areas that were previously impossible with standard cameras or inspection devices. The complete board is smaller than a business card, yet is a full function camera with built-in lens and automatic light compensation. The camera operates on 12 VDC and provides a quality picture on a standard TV monitor or VCR. The unit can also be connected to a computer.

The camera can be built into machines, assembly lines, robotic arms, isolated chambers, gauges, moving vehicles, and any application where remote vision can solve problems or improve the efficiency of an operation.

The camera comes with a wide-angle, 3.6-mm lens. The lens can be focused as close as 1/2 inch from an object, or from 1 foot to infinity. A selection of lenses is also available to suit a wide range of applications. A



modified version can be used with special microscope objectives for up to 300x magnification.

The camera is also infrared sensitive which allows viewing of nonvisible light sources such as laser beams or heating elements in the 800- to 1100-nm range. Picture elements are 540 pels horizontally by 492 pels vertically and the minimum resolution is 380 TV lines. Light sensitivity is 0.5 lux (0.05 foot-candles). The output conforms to EIA RS-170. A CCIR version is also available.

The module is 1.8" x 0.8" x 2.7". It uses a 3-pin modular plug for video, ground, and power. The V1206 can be run up to 1000' with new types of miniature coaxial cable.

This new-generation camera is considerably less costly (due to advanced LSI circuitry) than types originally used in the military, government, and scientific community. The V1206DC sells for \$189.

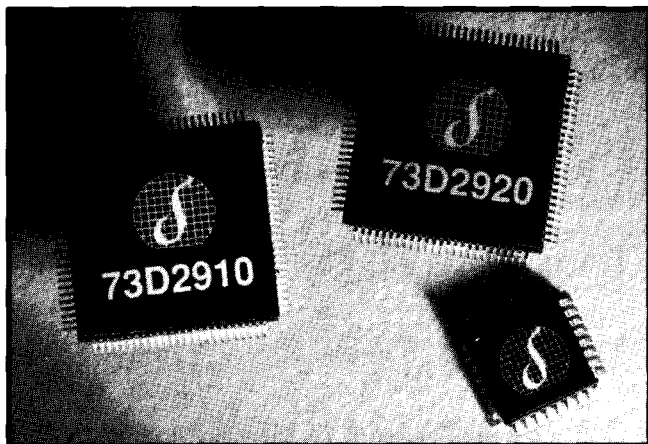
Optical Systems Div., Marshall Electronics Inc.
5649 Mesmer Ave. • Culver City, CA 90230
(310) 390-6608 • Fax: (310) 391-8926

#506

DATA/FAX/VOICE CHIP SET

Silicon Systems has announced the 73D2950, a data/fax/voice chip set that integrates a transformerless Data Access Arrangement (DAA) to provide a very low power, low-cost solution for fax modem designs. The optional 73D2950T version enables the designer to add high-quality voice annotation or audio files to data and fax transmission using a recently licensed audio compression algorithm.

The TrueSpeech algorithm developed by the DSP



Group Inc. enables a one-minute voice file to be stored in only 60K bytes of memory, versus as much as 940K bytes with other audio compression algorithms. This allows storage of nearly 25 minutes of audio on a single 3.5" floppy disk. The algorithm has also been selected by Microsoft for inclusion in the next version of Microsoft Windows Sound Systems.

The 73D2950/2950T includes three chips: the 73D2910 microcontroller, the 73D2920 DSP, and the 73D2930 analog front end line interface circuit (AFELIC). Smart power-down features are incorporated.

The 73D2910 is a custom microcontroller based on the industry-standard 80C52. The 73D2920T DSP option includes direct microphone input and direct speaker output. The 73D2930 incorporates the transformerless DAA and provides the necessary off-hook conditioning functions for connection of the chip set to both domestic and international public switched telephone networks.

The 73D2950/2950T data/fax/voice chip set sells in OEM quantities for \$22.00 and \$29.50, respectively.

Silicon Systems • 14351 Myford Rd. • Tustin, CA 92680
(714) 731-7110 • Fax: (714) 669-8814

#507

NEW PRODUCT NEWS

LIVE VIDEO IN PC WINDOW

Telebyte Technology has introduced the Model 722 PC Workmate, a device that allows live video windows to appear on a VGA display. The video window can be user-positioned or can occupy the entire VGA screen. This Picture-In-Picture (PIP) capability can be used for security surveillance, to view real-time data along with program applications, or at home performing computer-related tasks while watching the news or kids.

The Model 722 PC Workmate incorporates a full multiband VHF/UHF/CATV tuner which can receive up to 70 channels of TV information. In addition, the PC Workmate has direct video inputs for a VCR, video disc, camcorder, or surveillance camera and includes a built-in speaker. The unit is suitable for any standard VGA adapter and monitor, and delivers clear sound with high-resolution pictures.

It is connected between the PC system and the color display monitor. It merges the VGA signal with the

video signal and sends the combined signal to the VGA monitor. The video signal can come from the built-in tuner system or from an external video source.

Logically, the PC Workmate is transparent to the PC system. It can be treated as a TV set that shares the monitor with the PC system. It does not require any software and will not affect the PC operating system. Installation only requires moving cables. You can control display mode, PIP position, video source, channel, and more from the front panel of the PC Workmate.

The PC Workmate package is 15" x 3 3/8" x 10 1/2". Model 722 is supplied with all necessary mating cables. The unit is available as Model 722N to support the NTSC standard, or as Model 722P to support the PAL standard. Model 722 PC Workmate sells for \$525 each, with discounts for quantity.

Telebyte Technology, Inc.
270 E. Pulaski Rd. • Greenlawn, NY 11740
(516) 423-3232 • Fax: (516) 385-8184 #508

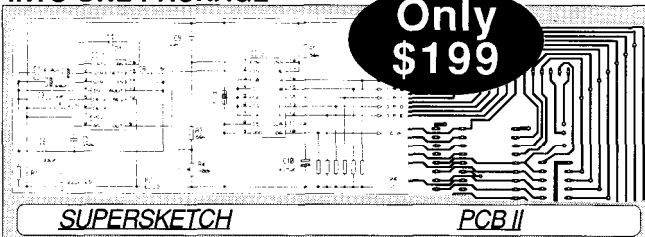


CADPAC II

TWO PROGRAMS FOR ONE LOW PRICE!!

SUPER SKETCH & PCB II: INTEGRATED INTO ONE PACKAGE

Only
\$199



PCB II & SUPER SKETCH features:

- * MOUSE DRIVEN *SUPPORTS CGA, EGA, VGA & SVGA,
- * OUTPUT TO 9 & 24 PIN PRINTERS, HP LASERJET & HPGL PLOTTERS. OUTPUT TO DTP PACKAGES *
- * PCB II ALSO HAS GERBER OUTPUT & VIEWING. *

THE EASIEST TO USE CAD AVAILABLE

R4 SYSTEMS Inc.

1111 Davis Drive, Suite 30-332
Newmarket, Ontario L3Y 7V1
(905) 898-0665
fax (905) 836-0274

Free DEMO Package
Write or Call Today

ALL PRICES ARE IN US FUNDS, PLEASE INCLUDE \$7 S/H

Download DEMO from BBS at (905) 898-0508 (2400/8/N/1)

Announcing

Robot Explorer™ Newsletter

Your guide to the world of mobile and intelligent robots.

With each new issue you'll descend into Antarctic volcanoes, design silicon ants, build solar-powered photovores and compete with automated firefighters. You'll also build the latest robot kits, review the newest books, and receive expert guidance from master robot crafters. Discover the fun and rewards of truly intelligent machines. Free your computers from the desktop and start them wandering about the floor.

Order #1 190 \$14.95 US/Canada/Mexico \$29.95 World-Wide

Mobile Robots

Inspiration to Implementation

MIT AI researchers Anita Flynn and Joseph Jones provide everything you need to start building robots today. Learn basic electrical and mechanical theory, microcomputer interfacing, and software techniques such as sub-summation. *Mobile Robots* won't leave you high and dry with a lot of theory. You'll find plans, schematics, and a rich array of resources. Start with a simple non-microprocessor robot and work up to a fully-intelligent machine. No other robot book is as detailed.

Order #9001:

Muscle Wire

Motion Without Motors!

Imagine having your own six-legged walking machine capable of tackling the most rugged-desktop terrains. The Muscle Wire kit provides a supply of Shape Memory Wire-thin filaments that contract when a voltage is applied. Muscle Wire is strong-wires the thickness of a human hair can lift copper pennies! In addition to sophisticated walking machines, the Muscle Wire project manual contains dozens of projects: from airplane launchers to animated butterflies. Projects for all ages and abilities.

Order #1071: \$59.95

Call Today

1.800.GO-ROBOT

(1.800.467.6268)

Mastercard/Visa Accepted

POB 458, Peterborough NH 03458

603.924.6079 (voice) 603.924.9441 (fax)

Internet orders accepted: apsol@world.std.com

FEATURES

14

Control Your Audio/Video Connections with the AVMux

24

Exploring the Vertical Blanking Interval

40

Employer Ownership of Employee and Consultant Work Product

Control Your Audio/Video Connections with the AVMux

One of Steve's more popular Circuit Cellar projects from years ago was the AVMux. With today's more sophisticated home theater requirements and the availability of newer, better chips, it's time to revisit the AVMux project.

FEATURE ARTICLE

Steve Ciarcia



about nine years ago, the concept of an entertainment (a.k.a., media) room

became a heavy topic in audiophile and videophile circles. Rec rooms lost their pool tables and pinball machines, and relatives expecting the same (free) spare bedroom were shown the Yellow Pages under "motels." Home builders who had just come to grips with jamming jacuzzis into the master bedroom closet now had to come up with a home theater to entice the sophisticated home buyer.

Aside from the physical room itself, the reality of a dedicated entertainment space was primarily a matter of assembling the components. Virtually all of the equipment (projection television, CD player, laserdisc, VCR, amps, etc.) was available off the shelf, but implementing the physical connections among the components presented a problem.

Virtually all of the equipment of sufficient quality for this purpose also had singular applications. Good stereo systems ("good" is interpreted as meaning audiophile quality) consisted of individual modules such as a tuner, preamplifier, amplifier, CD player, sound field processor, subwoofer amplifier, DAT player, and so forth. It was not uncommon to have to use eight or ten of these electronic subsystems to create the proper sound and video atmosphere in a true entertainment room.

The unfortunate truth back then was that while the concept of coordinated media was easily understood, an adequate means for channeling the audio and video signals among all

these subsystems that would ultimately result in the desired effect was no small task. In fact, it was a wiring maze. To ease the wiring congestion and simplify overall interconnection, I designed a combination 8x8 audio and video multiplexer.



Photo 1-The finished AVMux and hand-held control unit add both sophistication and class to any home theater installation.

The AVMux, as I called it, was (and is) exactly as the name implies. It was an electronic cross-matrix of switches which channels specific inputs to designated outputs. In the case of an 8x8 multiplexer, the inputs to it (commonly called the *froms*) come from the outputs of the various signal-generating subsystems, such as CD players, VCRs, laserdisc, tape recorders, and so forth (subsystems with both audio and video outputs would use the appropriate audio or video input side of the multiplexer). Similarly, the output side of the AVMux was connected to the inputs of various other system components (called *tos*): preamplifier, amplifiers, surround decoder, soundfield decoder, tape recorders, and so forth.

Using the AVMux and a series of push-button settings like "From 2 to 6" "From 3 to 5," and "From 3 to 7," would, for example, channel the laserdisc into a system combination of the preamp, amplifier, and subwoofer amp (we presume speakers are always connected to a particular amplifier). To change the source from the laserdisc to the CD player, we might enter "From 1 to 6" on the control. This "erases" the "From 2 to 6" connection and replaces it with "From 1 to 6" instead.

A SOLUTION NOW LACKING A PROBLEM?

Nine years later, the scenario is quite different. Consumer audio and video equipment have made significant advances such that the borderline between true "audiophile" and "high-end" consumer quality is fuzzy at best.

To obtain superb sound and video in the past, the only solution was to use more expensive modules and externally connect things through an electronic multiplexer like the AVMux.

Today, stereo manufacturers have succeeded in providing some real power and quality in a single enclosure. It is not hard to find the equivalent of the tuner, preamplifier, two 100-watt stereo amplifiers, subwoofer amplifier, a Pro-Logic surround decoder, and a dedicated audio/video multiplexer in one enclosure. A Mitsubishi unit I looked at recently included all this for \$1400. When I compare this to the \$10,000 I spent on the equivalent separate boxes.. .

AUX2 (remember, the tuner and surround decoder are already in the box, so they don't need external inputs), a simple push button or remote tells the internal multiplexer (frequently a relay) to channel the appropriate input through the preamp into the surround decoder and amplifiers. The outputs of this system are usually just speaker terminals and the direct video from the designated source. This video goes to a separate monitor such as a projection TV.

Depending upon the brand and the cost, you can find a variety of input/output configurations. Because they have a dedicated application, the audio multiplexer is always a Nx1 and the

The greatest attribute of today's "integrated component" single-box stereos [provided it is of sufficient quality] is that they contain an integral dedicated multiplexer. With external input designations like VCR1, VCR2, TAPE 1, TAPE2, CD, AUX1, and

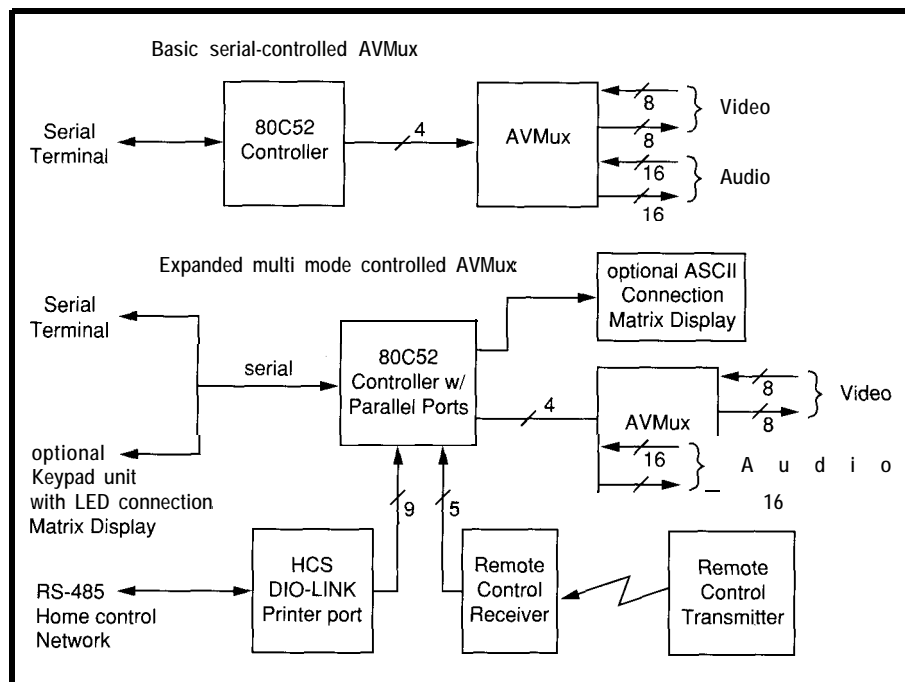


Figure 1-The AVMux setup can be as simple as a serial terminal controlling it, or as complex as using wireless controls, user displays, and the HCS II to make a truly powerful system.

video rarely more than a 4x1 or 4x2. The Nx1 audio configuration is because all the amplifiers are in the one box. Any sound created in the system has to be amplified by this box and all speakers are wired to it. If there are eight audio inputs and only a single output that goes to the amps, then we have an 8x1 audio mux. Similarly, since this integrated system does not incorporate its own monitor, the manufacturers facilitate an external connection

through the mux. With four video sources and two outputs, we have a 4x2 video mux equivalent.

OK, enough beating around the bush. If you want to put together an entertainment room and you are on a limited budget, these integrated systems are the only way to go. One hundred watts channeled through efficient speakers can sound quite good in any room.

The problem comes when you want to do something that is not part of the basic box. The instant you want to connect three video inputs where you only have two available, change the front surround speaker power output capability from 20 W per

channel to 120 W per channel, or, using the same CD player, channel its output to a different set of speakers.

While the integrated system has greatly reduced the introductory cost of an entertainment room and satisfied bottom-up multiplexer execution to a great extent, it has done little for applications that don't fit the "mold."

As you might already have guessed, very little that I do fits any "mold." Let me explain.

Recently I decided to revamp our entertainment room. I thought of using an integrated surround system like those I described, but even the 350-W Mitsubishi unit didn't quite fit the bill. While it contained all the

right elements, it didn't have anywhere enough power and its internal switch multiplexer (or that on virtually any integrated unit) was far too limited for the I/O selection I have.

In all truth, if I didn't already have all the amplifiers and many of the subsystem components, I might would have changed the rules to fit the solution, but the disparity in performance was still there. My attitude toward surround sound movies is to create an environment as close to the pictured event as possible. When we watch Top Gun, only the lack of exhaust fumes and salt spray keeps you from actually believing you are on a carrier deck. To achieve this sound envelope requires considerably more than 5 W through a 6" x 9" oval TV speaker. In my opinion, it also takes more than the 300-400 watts available from the current top-of-the-line integrated units.

My entertainment room technique involves using separate modules (preamp, surround decoder, etc.) instead of the integrated unit. The fact that a stand-alone surround decoder/sound field processor has all its signal outputs available, gives me the option of using any size amplifiers I want. Being brief, let me just say that I'm using about 1200 watts on the dozen speakers comprising the surround system. Included among them is a pair of 200-W subwoofers that guarantee you don't miss an F-14 fly by.

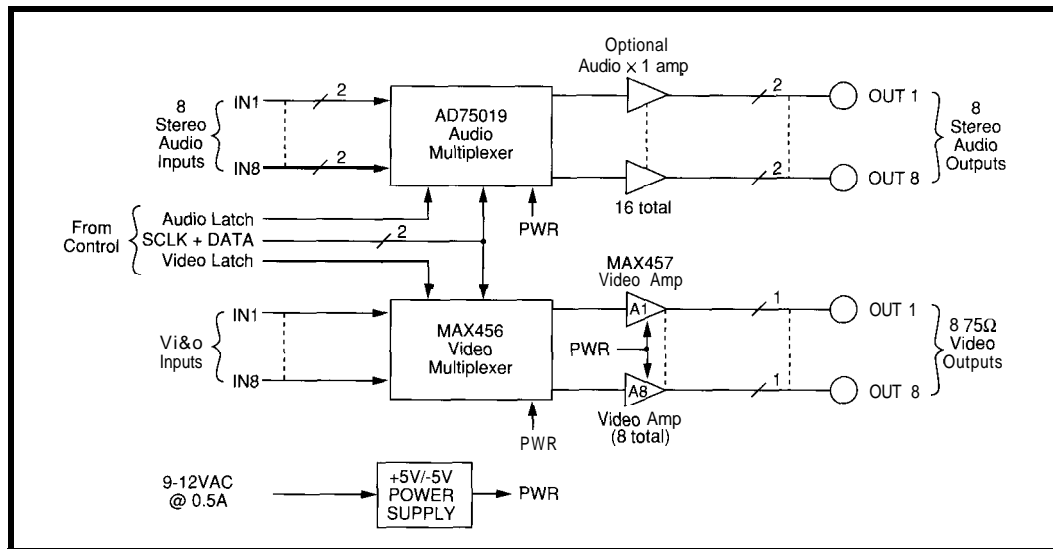


Figure 2—The Circuit Cellar VMux uses a pair of chips (one from Analog Devices and the other from Maxim) designed specifically for doing audio and video multiplexing.

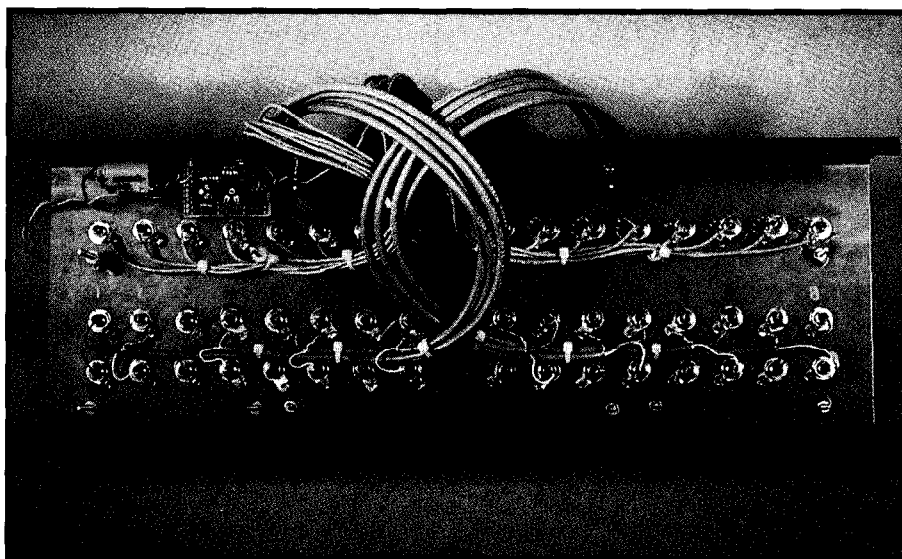


Photo 2—While the circuit isn't complex, wiring the project can be a major task since all the cables must be shielded.

What finally convinced me to redesign and further employ an external AVMux was the implementation of dual systems. I'm sure you've heard the saying, "If it ain't broke, don't fix it!" Well, that can apply to stereo systems as well.

Take away all the sound effects processing, decoding, and electronic signal manipulation and true sound quality of any stereo is simply an audio source, amplifier, and a pair of speakers. In my opinion, the sound produced by a pair of B&W 808

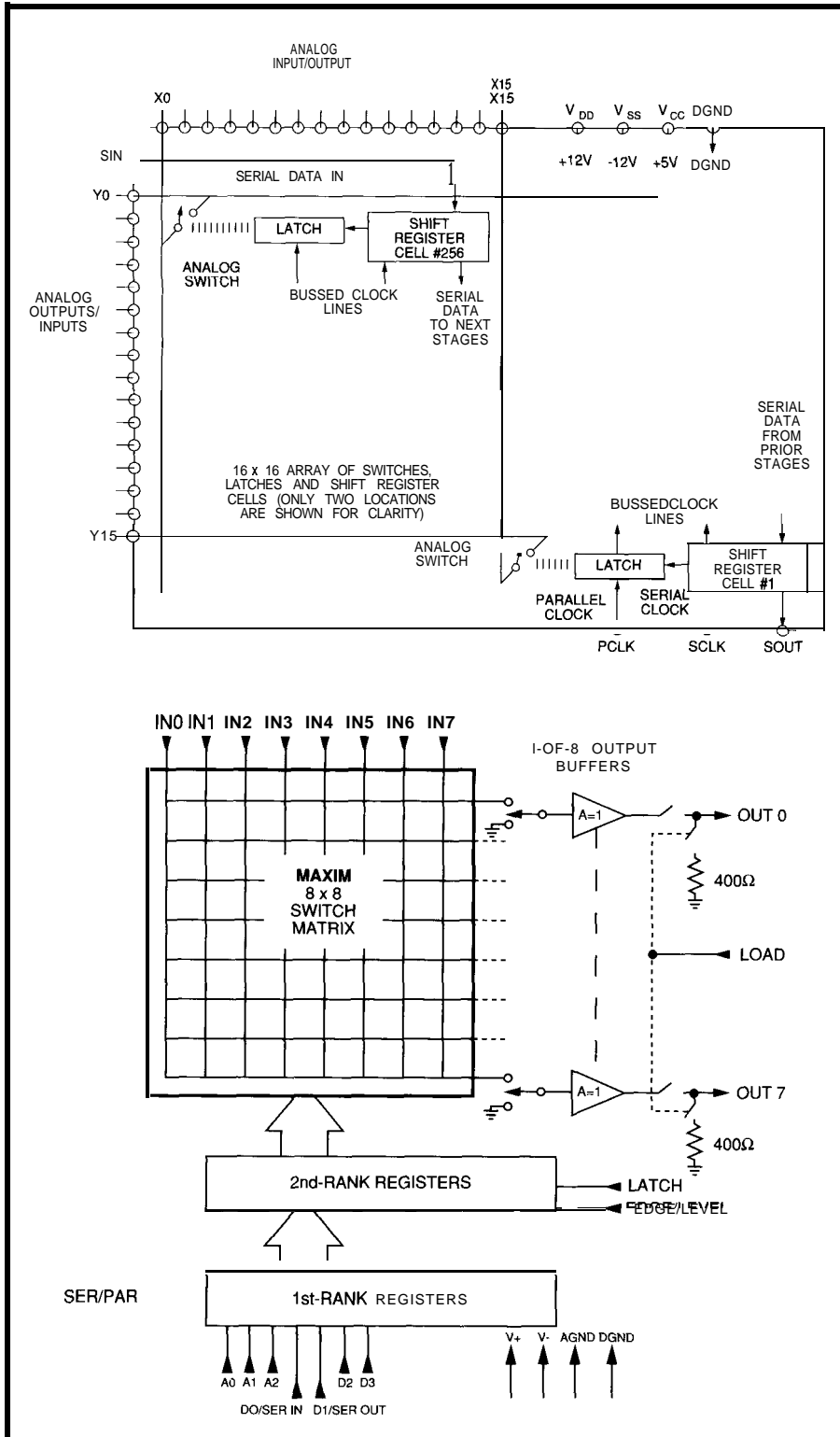
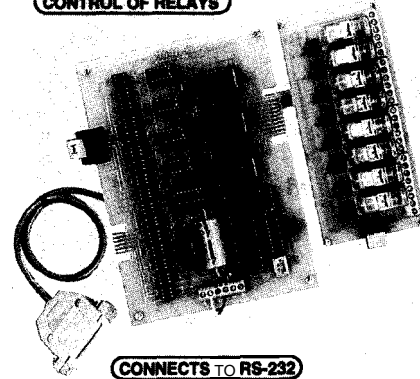


Figure 3-The Analog Devices AD75019 (top) is a 16x16 switch matrix ideal for use in audio applications while the Maxim MAX456 (bottom) is an 8x8 switch meant for video.

RELAY INTERFACE

PROVIDES SOFTWARE CONTROL OF RELAYS

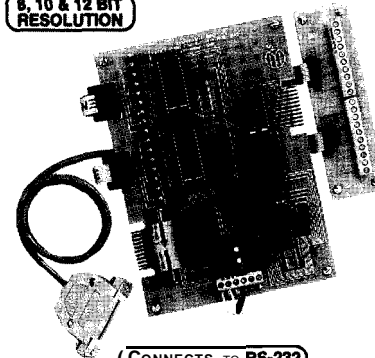


CONNECTS TO RS-232

AR-10 RELAY INTERFACE (16 channel).....\$ 89.95
 Two 8 channel (TTL level) outputs are provided for connection to relay cards or other devices (expandable to 128 relays using EX-16 expansion cards). A variety of relays cards and relays are stocked. Call for more info.
 AR-2 RELAY INTERFACE (2 relays, 10 amp).....\$ 44.95
 RD-8 REED RELAY CARD (8 relays, 10 VA).....\$ 49.95
 RH-8 RELAY CARD (10 amp SPDT, 277 VAC).....\$ 69.95

ANALOG TO DIGITAL

8, 10 & 12 BIT RESOLUTION



CONNECTS TO RS-232

ADC-16 AID CONVERTER* (16 channel/8 bit)...\$ 99.95
 ADC-8G AID CONVERTER* (8 channel/10 bit)...\$124.90
 Input voltage, amperage, pressure, energy *saga, joysticks and a wide variety of other types of analog signals. RS-422/RS-485 available (lengths to 4.W). Call for info on other A/D configurations and 12 bit converters (terminal block and cable sold separately).
 ADC-8E TEMPERATURE INTERFACE* (8 ch)...\$ 139.95
 Includes term. block & 8 temp. sensors (-40' to 146' F).
 STA-8 DIGITAL INTERFACE* (8 channel).....\$ 99.95
 Input on/off status of relays, switches, HVAC equipment, security devices, smoke detectors, and other devices.
 STA-8D TOUCH TONE INTERFACE*.....\$ 134.90
 Allows callers to select control functions from any phone.
 PS-4 PORT SELECTOR (4 channels RS-422).....\$ 79.95
 Converts an RS-232 port into 4 selectable RS-422 ports.
 CO-485 (RS-232 to RS-422/RS-485 converter).....\$ 44.95

● EXPANDABLE...expand your interface to control and monitor up to 512 relays, up to 576 digital inputs, up to 126 analog inputs or up to 126 temperature inputs using the PS-4, EX-16, ST-32 & AD-16 expansion cards.

FULL TECHNICAL SUPPORT...provided over the telephone by our staff. Technical reference & disk including test software & programming examples in Basic, C and assembly are provided with each order.

HIGH RELIABILITY...engineered for continuous 24 hour industrial applications with 10 years of proven performance in the energy management field.

CONNECTS TO RS-232, RS-422 or RS-485...use with IBM and compatibles. Mac and most computers. All standard baud rates and protocols (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

International & Domestic FAX (614) 464-9556
 Use for information, technical support & orders.
 ELECTRONIC ENERGY CONTROL, INC.
 360 South Fifth Street, Suite 604
 Columbus, Ohio 43215-5436

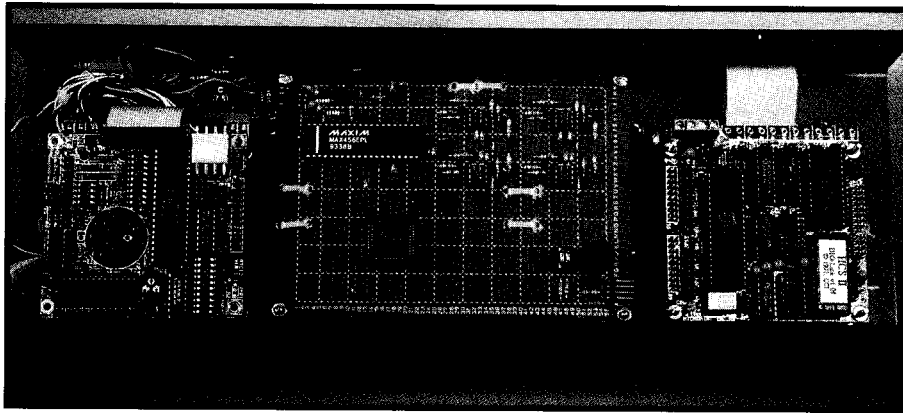


Photo 3—Inside the finished AVMux you can see the controller (left), multiplexer electronics (center), and HCS II DIO-Link (right).

Reference Speakers [running at another 1200 watts] comes close to perfect audio. I like classical music straight, while rock music seems better with all the effects. That means I have one system configuration for classical and a completely different one for rock.

Keeping a dual redundant system like this requires a sophisticated multiplexer to allow a common set of AV sources to be routed among all input combinations.

A SIMPLE MUX DESIGN WITH MANY OPTIONS

The new AVMux has improved performance primarily because it uses newer technology. Like the original unit, this is an 8x8 stereo audio and 8x8 video multiplexer. Unlike its predecessor, however, this mux allows the user to select independent or coordinated control of the audio and video sections. You can have eight audio sources with only three corresponding video signals and still use the other five video inputs for unrelated activities. Audio and video channels can be routed independently.

Due to its particular architecture, which allows independent audio and video operation, you can choose to make just a video-only or audio-only multiplexer. Control is realized with an 80C52-based microcontroller which allows considerable expansion options. As outlined in Figure 1, the new Circuit Cellar AVMux can operate through simple, direct, serial RS-232 control; operate with a wireless infrared remote control; operate with a hand-held keypad and LED connection matrix display; display an ASCII list of

connected points during operation; and allow network connection (using a DIO-Link) with functional control provided by the Circuit Cellar HCS II.

As you can see, this is a long list of functions. It also incorporates features that involve considerably more software than I cared to write. To finish in a reasonable time, I drafted Jeff Bachiochi to help. I'm glad I did and I'd further have to say that this new AVMux has as many features as it does because of his added expertise. Since we worked together, Jeff will be covering certain AVMux aspects in his column as well.

For this month, I'll describe the basic AVMux components and the controller design. Next month, Jeff will fill in more on how the software works, with particular emphasis on the remote keypad/LED connection matrix unit. I'll conclude next month with the wireless remote control interface specifics.

NEW SINGLE-CHIP MULTIPLEXERS

Any rational person would not even attempt to make an 8x8 multiplexer if it were not for the availability of some new highly integrated multiplexer chips from Analog Devices and Maxim. The configuration of the basic AVMux using these chips is outlined in Figure 2. Regardless of all the other optional features, this basic circuit stays the same. The two sections—audio and video—are also divisible and you could build each separately.

The audio multiplexer is the simpler of the two, both in concept and construction. Employing an

Analog Devices CMOS AD75019 16x16 crosspoint switch array (its functional block diagram is in Figure 3) allows the complete circuit to be fabricated with just that single chip.

Unlike its video companion, an 8x8 stereo audio multiplexer involves switching 16 signals (8 left channel and 8 right channel). I chose the AD75019 specifically so I would not have to use two 8x8 devices. The AD75019 contains 256 analog switches in a 16x16 array. Any of the X or Y pins may serve as inputs or outputs. Any of the X pins can be programmed to connect to any Y pins and vice versa. The switches allow amplitudes up to the supply voltages and have a typical resistance of 150 ohms.

Figure 4 shows the actual circuit connections. To keep things simple, we designated all X pins as inputs and all Y pins as outputs: X0-X7 are right channel inputs 1-8, X8-X15 are left channel inputs 1-8, Y0-Y7 are right channel outputs 1-8, and Y8-Y15 are left channel outputs 1-8. I know that chip designations use "0" instead of "1" as the first signal line, but try to tell someone unfamiliar with electronics that and you'll get a blank stare. Zero to seven is for the builder and one to eight is for the user.

Connecting input X2 to output Y7 is accomplished by closing the crosspoint switch at their intersection. Each of the 256 crosspoint switches is controlled by a shift register cell. If the cell contains a 0, the switch is open. If the cell contains a 1, the switch is closed. The contents of all the cells, describing the crosspoint matrix map, is a 256-bit data packet.

The data is loaded serially via the SIN input (DATA) and clocked into the 256-bit shift register via SCLK (CLK). When all 256 bits have been entered, data is transferred in parallel to the switch latches on the high-to-low leading edge transition of PCLK (ALD). One caution to note here: the serial shift register in the AD75019 is dynamic and the minimum clock rate is 20 kHz (the maximum rate is 5 MHz). With a high-level language controller, an assembly language routine is often used to send data to

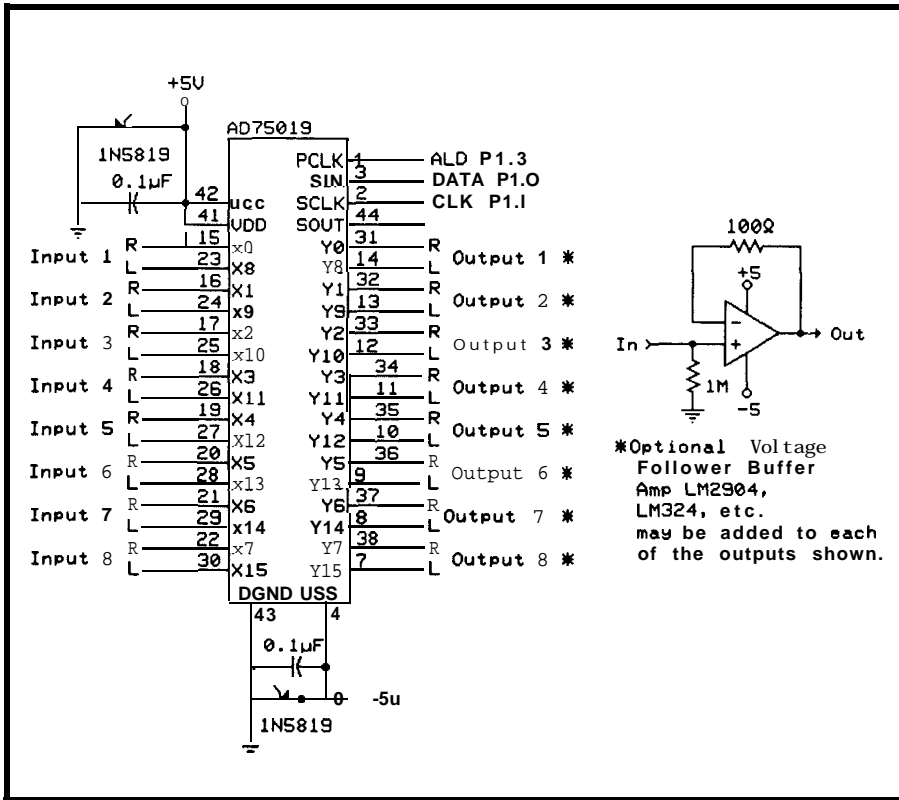


Figure 4—The actual connections to the AD75019 are numbered 1-8 rather than 0-7 to make it easier on the user.

this chip. Once programmed, however, operation is static and the switches stay programmed for as long as the power is on.

One final note on this chip. When powered on ± 5 V, the AD75019 has a typical switch resistance of 300 ohms. Many applications may be unaffected by this added resistance and you can use the circuit as is. If you are unsure or desire extra output protection, add the voltage follower buffer amplifier to each output channel (16 amps total) as noted in the schematic.

A NEW MAXIM MULTIPLEXER

If you look closely at the spec sheet on the AD75019, you'll note that it has a switch frequency response of 20 MHz. While I could have used it for both sections, I opted to use a chip specifically designed for video.

I decided to make the AVMUX video section using the new Maxim MAX456 CMOS 8x8 crosspoint switch (functional block diagram in Figure 3) specifically designed for video. It contains a similar register, latch, and switch arrangement as the audio mux, but the MAX456's connection logic prohibits illegal switch connections.

The MAX456 is an 8x8 matrix, and there are indeed 64 separate switches, but in video applications, not all crosspoint connections are allowed. Video outputs should not be tied together, for example.

In the MAX456, each output connection command is thought of as a combination of the binary address of the selected 1-of-8 inputs and data such as whether the buffer is on, off, or grounded. This data combination results in a 4-bit value for each output channel. To program 8 channels, therefore, only requires 32 bits.

The MAX456 is unique in that it can load this switch register data by either parallel or serial means. To keep programming consistency, I chose the latter. The 32 bits are applied to the DO/SER IN (DATA) line as the \bullet WR (CLK) line is clocked (these are the same two lines that are connected to the audio mux). At its conclusion, the data is latched on a high-to-low leading-edge transition of the LATCH pin (VLD). Using common data and clock control lines with separate video load (VLD) and audio load (ALD) allows just four control lines to control the whole AVMUX.

\$149 ICE?!!!

Yes, that's right! HTE has dropped the price of its popular 8031/32 Enhanced DryICE from \$269 to \$149. Now you can't afford to be without one! This ICE performs single step, real-time execute to breakpoint, disassembly and more. 8K user code space, expandable to 32K. Order yours today!

More ICE

Our DryICE Plus has so many features, the price is unbelievable. 48K of user code space, real-time execution, and expandability to nearly all of the 8051 family processors. We are now supporting pods for the 803112, 8751, 80CL51/32, 8051Fx, 80C154, 80C410/1, 80C451, 80C528, 80C550, 80C535, 80C537, 80C552/562, 80C575, 80C652, 80CL781/2, and 80C851. With pod\$ priced at only \$149, going to a different, higher integration processor is easy. Pod and base unit are just \$446 complete!

Controller 80C552

At \$149, our 552SBC-10 OEM board has the price and features you need right now! It's an 8051 core processor with an eight channel, 10-bit A/D, two PWM outputs, capture/compare registers, 16 digital I/O, one RS232 serial port, four JEDEC memory sockets. Add options like two more RS232/422/485 ports, 24 more digital I/O, real-time clock, serial EEPROM, and battery-backup for clock and RAM. Start with the Development board; all the peripherals and a debug monitor for only \$349. Download and debug your code, assembly or 'C', right on the SBC, then use the low-cost OEM board of your choice for production. We also do customs - call for a free quotation.

80C188 SBC

MEW! Starting at only \$249. Two serial RS232/422/485, Parallel, expand to 16 ch. 12-bit A/D, 8 ch. 12 bit D/A, Keybd, _CD, Relay I/F, more. Call for details!

Call for your custom product needs. Quick Response



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

(619) 566-1 892

70662. 1241 @compuserve.com

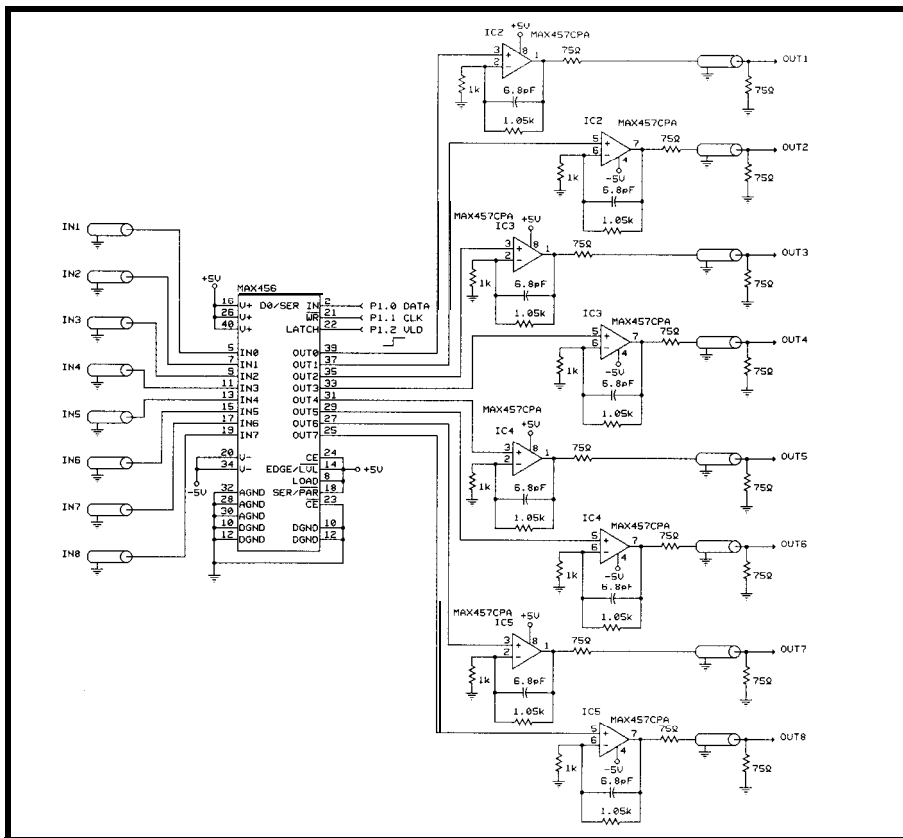


Figure 5—While the circuitry around the MAX456 video mux is simple, having to use shielded cable complicates wiring. The MAX457 video amps are used for impedance matching.

One additional note on the MAX456. While it does have internal video buffers, they can only drive 400-ohm loads. Since most video applications require 75-ohm impedance, additional external buffer amplifiers have to be added. The dual-amp MAX457, which has a 70-MHz unity-gain bandwidth, is the perfect choice. As connected in Figure 5, the MAX457 is configured for a closed-loop gain of 2. The gain selection resistor is set at 1.05k instead of 1 k to make up for open-loop gain loss. The 6.8-pF capacitor helps eliminate phase delay at high frequencies.

We have to be careful to properly match the cable impedances even in a hand-wired prototype. This particular circuit allows us to drive a doubly terminated 75-ohm line. The combination of the series 75-ohm resistor and the 75-ohm termination at the output connector produces a low-noise 75-ohm external connection with the appearance of unity-gain amplification.

A FAMILIAR CONTROLLER FOR THE AVMUX

As I mentioned, it takes just 4 wires from the controller to the AVMux chips to do everything. Figure 6b is an 80C52-BASIC-based microcontroller circuit which is more than adequate for the task. Together, they represent the configuration outlined in Figure 1. I used a Micromint RTC52 and RTCIO as the controller rather than spending the time and effort to build one. At the price I get them, it wasn't a hard choice.

We chose to use BASIC (with assembly language calls) here because this is not a speed-intensive control application. Using a high-level language also expedited development and made certain expansion options easier. For example, since BASIC-52 supports a serial printer output, presenting an ASCII connection list for terminal display was as easy as adding an LPRINT (PRINT#) command.

The controller circuit has two sections: basic controller with one serial port and one parallel port (Figure 6a), and the same basic configuration with more parallel ports and nonvola-

Video Frame Grabber

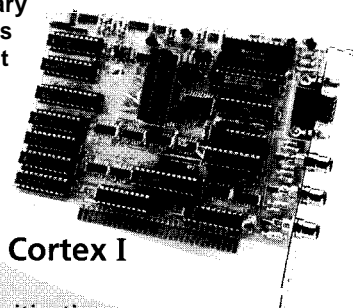
- \$495 Including Software with "C" Library
- Half Slot Card for Compact Applications
- Real Time Imaging with Display Output
- 8 Bit (256 Gray Levels)

FEATURES:

- Single 512 X 484 Image or Four 256 X 242 Images
- External Trigger
- Input Look Up Table
- Low Power Option Available
- Elegant Software Interface
- <10 nsec Pixel Jitter Means Accurate Digitization
- EISA (PC) Bus and STD Bus Products Available
- RS-170 and CCIR Video Formats Available
- Binary and TIFF File Formats

The Cortex I is used in machine vision, industrial control, medical, security and scientific applications around the world.

ImageNation strives to delight customers with quality products and personal service at a competitive price.



Cortex I

Call today for volume pricing or to discuss your application.

ImageNation Corporation
Providing Imaging Solutions
FAX (503) 643-2458 • (800) 366-9131

P.O. Box 276
Beaverton, OR 97075
(503) 641-7408

tile memory (Figures 6a and 6b). Since this circuit has been presented on numerous occasions, I will not belabor it. The initial circuit is suitable for operating the AVMux through a serial terminal. A simple connection display and command menu prompts the user to designate the desired From-To connections. More later.

PROTOTYPING

Success here is not just a function of interpreting the schematic correctly. Experience counts.

As you can see from the photos, even though this is only a six-chip circuit, it is a wiring nightmare. To keep signals from radiating or picking up the radiation from adjacent signals, all input and output wires should be shielded cable [now you might understand the reason for doubly terminated video buffers). I even used shielded cable from the MAX456 to the buffer amps. The enclosure should be metal to serve both as an EM1 shield and as a large noise-reducing common ground. It can also function as the heatsink for your power supply.

Ordinarily, I wouldn't discuss the power supply in any detail. In an application where limiting electrical noise is critical, I believe it a noteworthy exception. First of all, I suggest

you use a linear supply only and not a switching regulated unit. Without the benefits of printed circuit board layout, ground planes, single-point termination, and proper shielding (all of which are incorporated in a production unit), a switching power supply is just another source of unwanted noise (herringbone pattern on the video and buzz in the audio).

Getting from here to there can be an interesting journey, however. I designed my AVMux as a totally enclosed unit. My intention was to use one of those cheap wall-module power sources and an internal regulator. The AVMux operates on +5 V and -5 V. Unfortunately, all the cheap wall-modules seemed to be 2-wire output.

Since I couldn't just throw in a switching regulator or charge pump inverter to get -5 V, I concluded that my only option was an elaborate dual tracking regulator with a concocted ground. Would I have to resort to an external power supply and pipe everything in instead?

The answer turned out to be really low tech. Shown in Figure 7, the AVMux power supply is just a regulated-output AC-to-DC converter. We can get by with a 2-wire voltage source if it is AC. The AC is then half-wave rectified and stored as separate positive

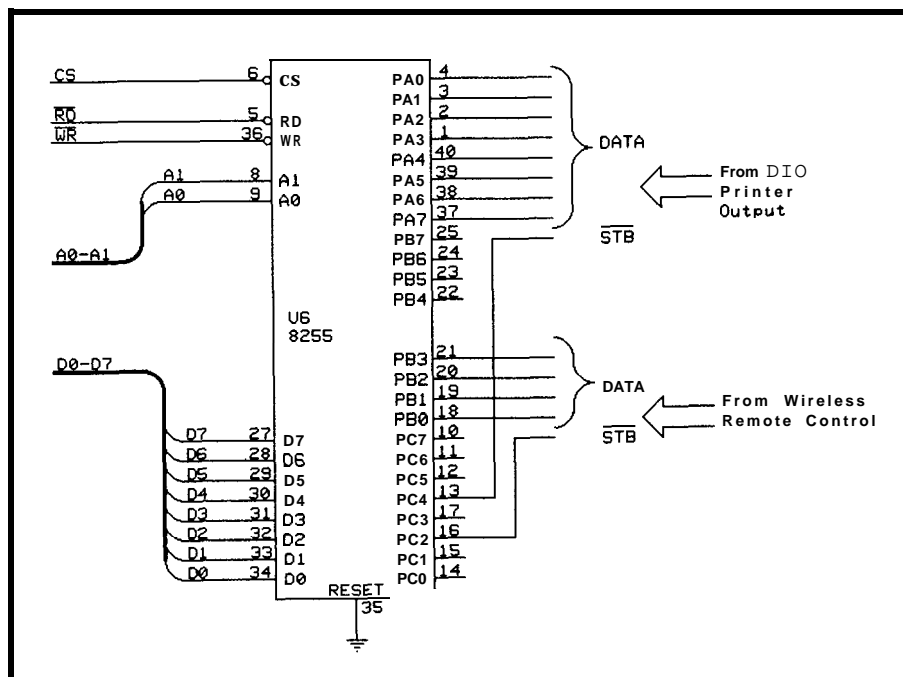
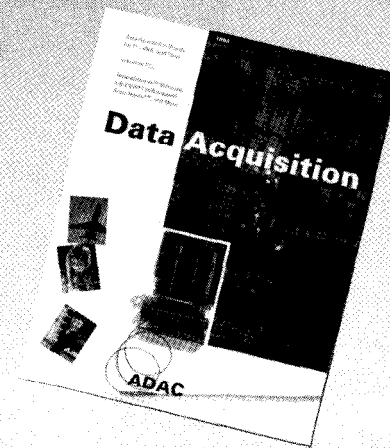


Figure 6a—To add either HCS II or wireless remote control to the AVMux, an 82C.55 must be added to the core controller in Figure 6b.

NEW Data Acquisition Catalog

Covers expanded line.



FREE!

1994 120 page catalog for PC, VME, and Qbus data acquisition. Plus informative application notes regarding anti-alias filtering, signal conditioning, and more.

NEW Software:
**LabVIEW[®], LabWindows[®],
Snap-Master[™]**, and more

NEW Low Cost I/O Boards

NEW Industrial PCs

NEW **Isolated** Analog and Digital Industrial I/O

New from the inventors of plug-in data acquisition.

Call, fax, or **mail** for your free copy today.

ADAC

American Data Acquisition Corporation
70 Tower Office Park, Woburn, MA 01801
Phone: (800) 648-6589 Fax: (617) 938-6553

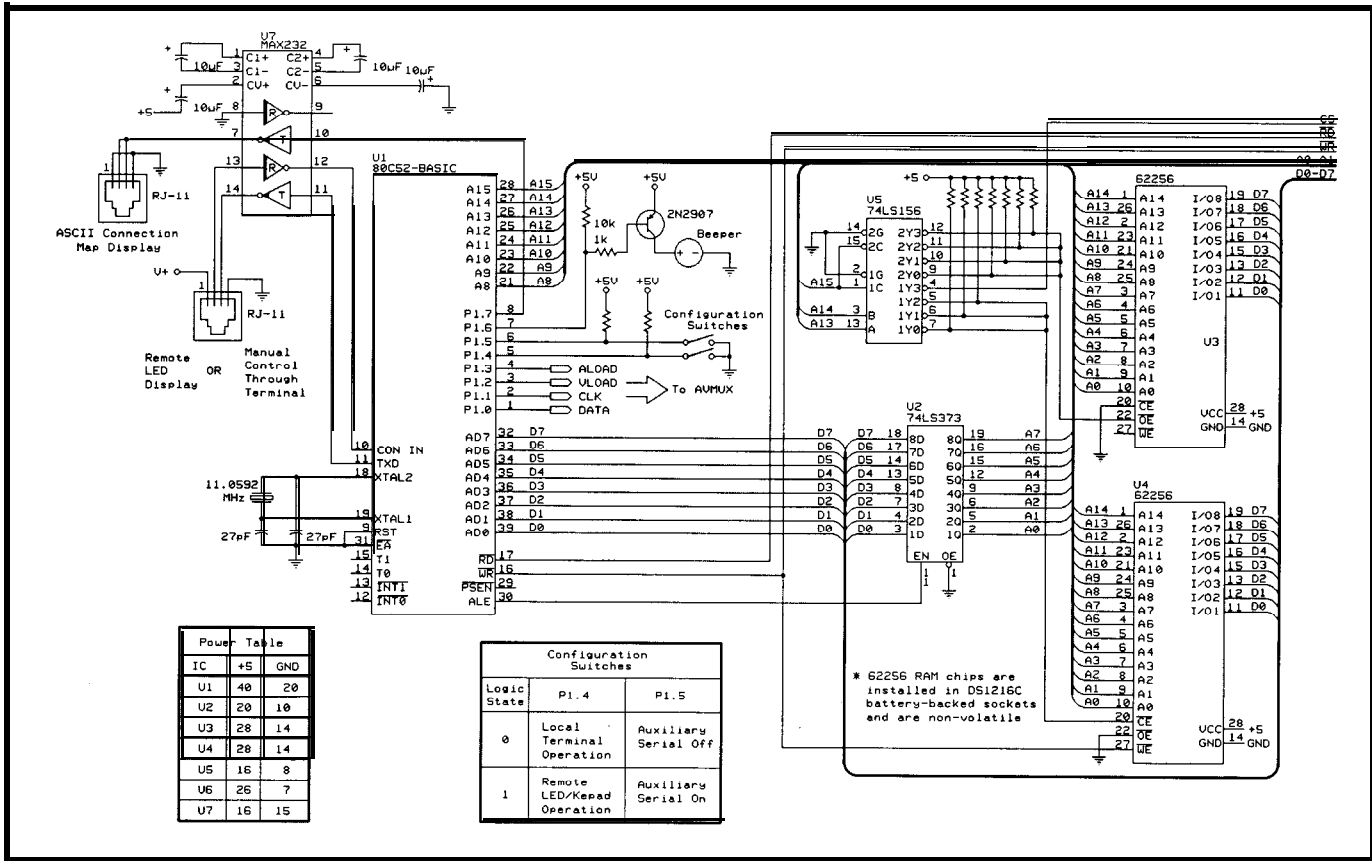


Figure 6b—The control section of the AVMux uses the same 80C52-BASIC core seen in many other projects. Connections to the outside world are done through RJ11 jacks

and negative supplies with a common ground. Two three-terminal regulators simply convert this to +5 V and -5 V, respectively. The one notable circuit peculiarity is that each regulator has been configured for voltage output adjustment with a pot. By referencing these pots to the opposite supplies, the overall input-output differential necessary for these three-terminal regulators is reduced. Rather than a 12-VAC input, a 9-VAC will work just fine. The benefit is reduced power dissipation and a cooler AVMux box.

AVMUX OPERATION

The AVMux is designed to operate in a variety of modes. With just the basic serial controller, all user interaction is through a terminal. When initialized, the program displays a connection matrix. To set a connection, the user designates whether it is (A)udio, (V)ideo, or (B)oth. Answering (A) displays the audio matrix while (V) or (B) displays the video matrix. The next entries are the From and To channel numbers, or cancel. The program then repeats the display,

showing it with the new connections (Jeff's program is on the Circuit Cellar BBS for anyone building the AVMux).

As I mentioned, by using BASIC we have a serial printer output also available. One program function physically documents the entire From-To connection list, by component names (such as From ADS CD Player To Nakamichi Preamp Aux 2 input). This is for people who don't want to carry a component list to remember channel numbers or for those who want to have space in their stereo cabinet and want a unique terminal display.

EXPANDING AVMUX FEATURES—THE HCS CONNECTION

Adding the 8255 PPI to the basic controller circuit facilitates other AVMux features. Initial operation is through a serial ASCII terminal. Considering that the communication is simply the 12 characters typically on a telephone keypad, other methods can be employed to communicate these characters.

The first option we thought of was a hand-held remote that presented a display of the connected channels on an 8x8 LED array. It also had an integral keypad to enter commands.

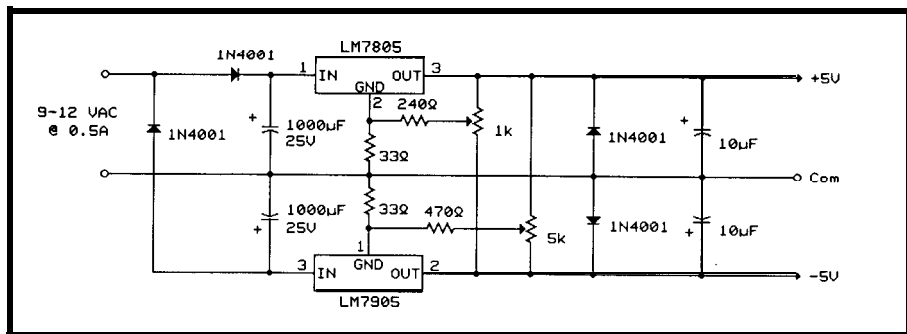


Figure 7—The power supply section creates quiet positive and negative supplies for the amplifiers from an AC source.


RICE16-5x

This construction of this unit and a description of its software will be presented by Jeff next month.

The next brain storm we had was to connect the AVMux so it could work independently from, or in conjunction with, the HCS II and the trainable infrared MCIR-Link. The connection is accomplished using the DIO-Link.

Normally, the DIO-Link is used as an 8-bit, bit-programmable I/O port on the HCS II's RS-485 network. The DIO-Link can also be set to drive a parallel printer. In this configuration, all bits are outputs and each character transfer is accompanied by a strobe pulse. Character transfer timing is programmable within the DIO-Link.

Port A and one bit of Port C are used to receive the parallel data. The HCS can command the AVMux by simply sending an ASCII string, such as "A35" (make an Audio connection from 3 to 5), to the DIO-Link printer port. It's as easy as that.

Since we were accepting parallel data inputs anyway, it didn't seem outrageous to consider an additional source. The total 12-character vocabulary could also be represented as 12 combinations within a 4-bit code. It shouldn't be too hard to communicate these codes using readily available remote control transceiver chips. In fact, next month I'll demonstrate such a wireless remote interface design. 

Steve Ciarcia is an electronics engineer and computer consultant with experience in process control, digital design, and product development. He may be reached at steve.ciarcia@circellar.com.

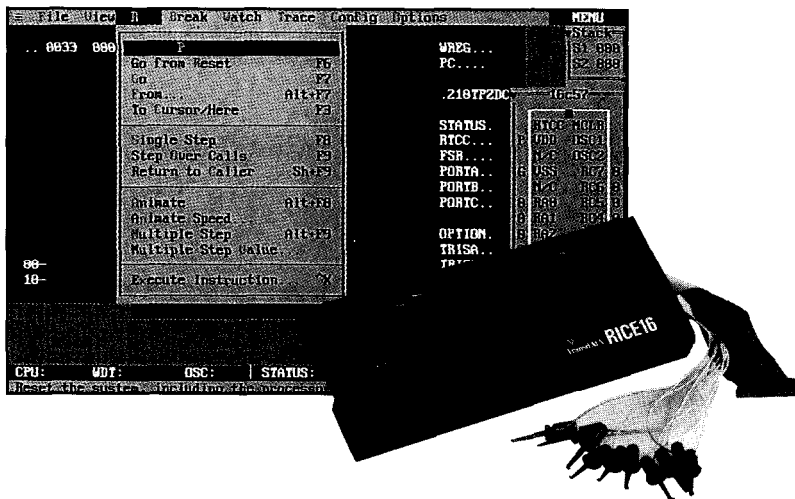
SOURCE

Some components of this project may be purchased from

Pure Unobtainium
13109 Old Creedmoor Rd.
Raleigh, NC 27613
Phone/fax: (919) 676-4525

IRS

401 Very Useful
402 Moderately Useful
403 Not Useful



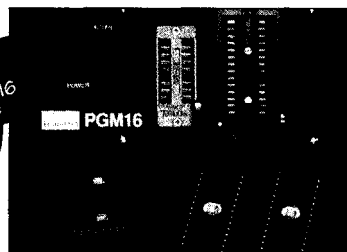
Real-Time Emulation to 16MHz

Complete system for **16C5x** only \$895.00'

Introducing RICE16-5x, real-time, non-intrusive in-circuit emulator for the PIC16C5x family microcontrollers: a feature-filled development system at an affordable price. * Suggested Retail for US only

- Real-time Emulation to 16MHz
- Non-intrusive Operation
- PC-Hosted via Parallel Port
- 8K Program Memory
- 8K deep by 24 bits wide Trace Memory
- Source Level Debugging
- Unlimited Breakpoints
- External Trigger Break with either "AND/OR" with Breakpoints
- Trigger Outputs on any Address Range
- 12 External Logic Probes
- User-Selectable Internal Clock from 40 frequencies or External Clock
- Easy-to-use windowed software
- Single Step, Multiple Step, To Cursor, Step over Call, Return to Caller, etc.
- Search Capability in Source Code, Program Memory and Trace Buffer
- On-line Assembler for Patching Instruction
- Support; 16C71 and 16C84 with Optional Interchangeable Probe Cards
- Comes Complete with TASM16 Macro Assembler, Windowed Software, Power Adapter, Parallel Adapter Cable and User's Guide
- 30-day Money Back Guarantee
- Made in the U.S.A.

Call our BBS at (214)980-0067 to download **DEMO:RICE16.ZIP**



* for a limited time only



Advanced **Transdata** Corporation Tel (214) 980-2960
14330 Midway Road, Suite 104, Dallas, Texas 75244 Fax (214) 980-2937



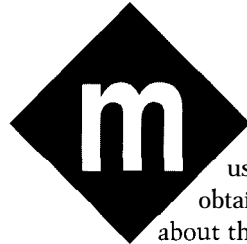
FEATURE ARTICLE

Mike Barnes

Exploring the Vertical Blanking Interval

A Tool for Gathering Data from your TV

Have you ever wondered what useful information might be hiding between the lines of your typical mindless television show? Mike did, so pulled out his soldering iron and cross-assembler to find out.



Most of us who use computers to obtain information about the outside world use modems. However, there are alternative methods, one of the more creative of which is already present in your house in the form of a television. If you're interested in a fairly accurate, free time-source for updating computer clocks, as well as a wealth of news and information to read at leisure from your terminal, you should take a second look at TV. Not watching it, reading it. Transcripts of many of your favorite TV shows, and several other interesting services are sent daily in a part of the television picture you never see unless your vertical hold doesn't work—because they are “hidden” in the vertical blanking interval.

The *vertical blanking interval*, or *VBI*, is simply the first 21 lines of each video frame. It is the portion of the picture not normally seen unless you adjust the vertical hold control so the picture rolls. Since the first nine lines of the VBI are equalizing and vertical sync pulses, only lines 10–21 contain useful data.

What kinds of things are found in the VBI? The original and still most

prevalent signals are calibration signals such as the *Vertical Interval Test Signal* (VITS) and *Vertical Interval Reference Signal* (VIRS). While important for measurement purposes, they are otherwise fairly boring. The more interesting signals fall into the categories of teletext, closed captions, and time stamps.

Teletext is the sending of text and/or graphic information via television. It is a subset of *videotext*, which is an interactive two-way system usually transmitted over telephone lines. Because teletext is sent via TV, it is noninteractive in nature. Teletext systems require the receiver of the information to use a special decoder for display. Most teletext systems send information in *pages* with the user requesting a specific page to be displayed. Because teletext systems were developed at a time when memory was expensive, teletext pages are sent over and over so the decoder need only have enough memory to store one page. The penalty for this is you must wait for each page after you request it. The maximum number of pages in most systems is 100-200.

The most common form of teletext in the U.S. is *closed captioning*. Closed captions allow millions of hearing impaired viewers to enjoy TV through the use of a decoder which displays the ASCII captions sent in the VBI. The other digitally encoded signal found in the VBI of the three major networks is a *time stamp* containing the current time and date.

In addition to describing the ways and means of teletext signals, I will describe the *VBI Explorer*, an 8031-based project allowing you to explore the VBI and its various data signals. It lets you select which VBI lines to monitor, and generates sync and video signals for display on an oscilloscope. Additionally, it can display raw *World Standard Teletext* or *North American Basic Teletext* signals and will also decode pages and display them on a CRT, or provide an interface for transmission to a computer. Closed captions can be displayed in a variety of forms ranging from raw data to fully decoded transcripts. Both caption and

text channels are supported in closed caption mode. Network time stamp signals can be displayed either in raw or decoded mode. The project can be used either stand-alone or connected to a computer.

The VBI Explorer is designed to be the data gathering unit of a personal information system. In such a system, the VBI Explorer feeds data to a PC. Your favorite network and often local news programs are automatically stored for reading at your leisure. You can even have the system flag keywords and notify you of items of potential interest. Not only

news, but programs such as *Today*, *Good Morning America*, *Nova*, *Washington Week in Review*, *Meet the Press*, *Tonight Show*, and countless others are captioned with enough clues to detect speaker changes and generate very readable transcripts. In addition to gathering closed caption information, teletext magazines can be stored and accessed without having to wait for the next page to roll around.

Such a concept is not new. The M.I.T. Media Lab has been working on this concept for many years, under various projects such as *Newsprint*, which stores captions along with an occasional digitized picture, and *Personal Newspaper*, which uses captions along with other material to present information customized to the individual user.

This article assumes you have a basic understanding of video signals, including such concepts as *fields*, *frames*, *blanking intervals*, and *front porches*. If these are unfamiliar, overviews of the video signal can be found in many electronics magazines as well as library books.

WHAT'S AVAILABLE?

While there are many different teletext formats in existence, three types are commonly found in the U.S.,

ranging from the slow and archaic to the fast and complicated. The slowest of these is the *line 21 captioning system*, commonly referred to as "closed captioning" since it is primarily used to transmit captions for hearing impaired viewers. The slowness comes from sending only two 7-bit, odd parity characters in field 1 of

graphics and color capability is built into the system by using a *mosaic character set*. Unfortunately, most of the WST signals are being phased out. In 1993 cable superstation WTBS eliminated its long-standing teletext magazine.

The Cadillac of teletext is the *North American Basic Teletext*

Specification

(NABTS). There are actually two distinct pieces to this system—the actual teletext backbone and the data contained within it. Technically, "NABTS" refers only to the way the teletext data is structured on the VBI line and

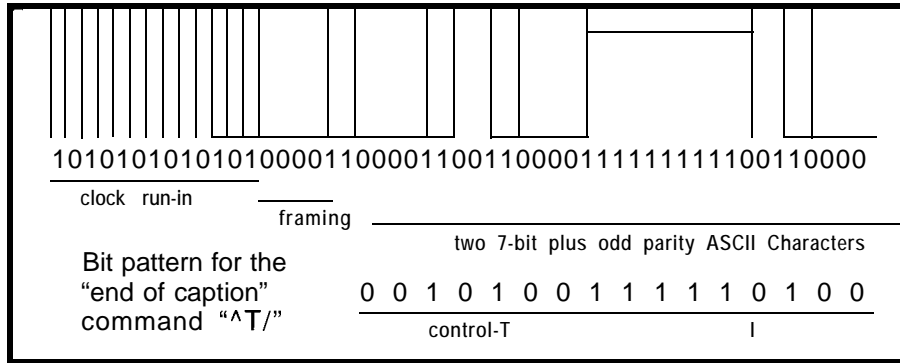


Figure 1—The closed captioning signal, usually found in line 21 of the vertical blanking interval, consists of a clock run-in, framing, and two ASCII characters per frame.

video line 21, resulting in a maximum of 60 characters per second. Even though the throughput is low, it is adequate for transcription purposes.

The data stream consists of raw text interspersed with commands dividing the stream into two time-multiplexed caption channels and two 15-line by 32-character text channels. However, newer decoders are capable of receiving only one caption and one text channel, so it appears the other channel is being phased out. Since 1979, many programs have been closed captioned, and the list expands each year. All major network news shows are captioned, as well as most prime-time and many PBS programs.

World System Teletext (WST) is the medium complexity teletext service. It received its origins from the British *ORACLE system*, which has been broadcasting since 1974 and has undergone several revisions since then. Data is grouped into *pages* of 24 lines of 40 characters. The American version sends only 34 characters per VBI line as opposed to 40 in the British system. Page size in the American system is still 24 by 40 using a special technique called *gearing*. Data may be sent on one or more VBI lines, with each line containing 34 7-bit, odd parity, ASCII characters. Limited

how packets of data are grouped. The actual data is encoded using the *North American Presentation Level Protocol Syntax* (NAPLPS, pronounced "nap-lips"). NAPLPS is a graphical language which describes color, shapes, and scaling information for the production of a two-dimensional picture. There really is no comparison between the mosaic images of WST and the alpha-geometric NAPLPS images—NAPLPS is the hands-down winner. The price for all this sophistication is an increased difficulty of decoding and displaying the images. The teletext data for NABTS, like WST, can be sent on multiple VBI lines, however NABTS sends 33 characters to a line instead of 34. At one time CBS was sending a three VBI line NABTS/NAPLPS news, weather, and sports magazine known as *Extravision*, but it is no longer present.

Last but not least is some interesting data which isn't teletext at all! Each of the three major networks sends a time stamp during field one of VBI line 20. This code consists of the month, day, hour, minute, second, and fraction of a second (in 1/30 second increments). During direct network feeds, the data is accurate to within a second or so of the correct time (from WWV, the national time standard),

Preamble Codes

Control Char	Printina Char	Row Num	abcd	Indent	Color	abcd	Indent	Color	
^Q	0010001	10abcde	1	0000	0	White	1000	0	White
	0010001	11abcde	2	0001	0	Green	1001	4	White
^R	0010010	10abcde	3	0010	0	Blue	1010	8	White
	0010010	11abcde	4	0011	0	Cyan	1011	12	White
^S	0010011	10abcde	12	0100	0	Red	1100	16	White
	0010011	11abcde	13	0101	0	Yellow	1101	20	White
^T	0010100	10abcde	14	0110	0	Magenta	1110	24	White
	0010100	11abcde	15	0111	0	Italics	1111	28	White

Midcaption Codes, Special Characters, and Control Codes

Control Char	Printina Char	Action	Control Char	Printina Char	Action		
^Q	0010001	01000e	^T	0010100	<space> 0100000	Resumecapti onl oading	
		010001e		"	0100010	Alarm Off	
		010010e		#	0100011	Alarm On	
		010011e		%	0100101	Roll-up mode (2 rows)	
		010100e		&	0100110	Roll-up mode(3 rows)	
		010101e		,	0100111	Roll-up mode(4 rows)	
		010110e		/	0101100	Erase Di splay	
		010111e		/	0101111	End of caption	
	0	0110000		¼	+	0101011	Resumetextmodel oading
	1	0110001		/(slash)	-	0101101	Carriage Return
	2	0110010		½			
3	0110011	¾					
4	0110100	⊘					
6	0110101	£					
7	0110111	musical note					

If "e" is 1, the caption is underlined.

Table 1—The closed captioning control codes allow for positioning, special characters, simple effects, and color, though the color isn't often used.

making it an ideal way to periodically set your computer's clock. You must beware, however, to take it seriously only during a nondelayed network feed. Tape-delayed programs, as well as other shows, will have a time stamp containing the incorrect time. Most times are sent as Eastern Local Time.

While the VBI Explorer explicitly supports closed captioning, network time stamping, WST, and NABTS, I don't have space to get into the details of latter two. See the reference section at the end of the article for more on these and other VBI signals.

CLOSED CAPTIONS-DECODING THE SIGNAL

One of the earliest forms of teletext is also the most commonly used in the United States. Closed captions are primarily used to provide captions for the hearing impaired, although the system was originally envisioned to be a complete teletext system, with two caption and two text channels. The second caption channel was designed for bilingual captioning, but this feature has gone largely unused. Today, only the primary

caption and text channels are used. For the last three years, all network prime time programming, as well as network news, sports, and many local newscasts have become captioned. The popularity of captioning is likely to increase since all televisions 13" or larger manufactured after July 1, 1993 must be able to display captions.

There are two basic types of captions—live or taped. *Taped captions* are produced by an editing process prior to broadcast. A half-hour taped program requires about 15 hours to create the captions. Taped captions are essentially transcripts of the program, with text placed under or above the person speaking. They tend to be very accurate representations of the spoken word. The same is not quite as true for live captions.

For live news and sporting events, presidential addresses and the like, the captions must be created on the spot. Known as *real-time captioning*, a stenographer listens to the speaker and keys in the phonetics in a manner similar to a court reporter. A computer program then evaluates the phonetics and produces the text which is then

broadcast with a lag of five to ten seconds. The system works remarkably well, though it suffers from phonetically based transcription errors. For example, "Iran" may become "Iran" and "President Reagan" might transcribe as "pressed I dent ray gun."

A variation of real-time captioning is used by many of the over 80 local stations that caption their news. Called electronic newsroom, the scripted portions of the newscast sent to the teleprompters are displayed as captions. Unscripted portions, such as on the scene reports, are not captioned.

For a technical description of how closed captions work, refer to Figure 1. Captions are normally sent on line 21 of the VBI in field 1, though they may rarely be found in other places. The signal consists of seven cycles of "clock run-in," followed by a start bit and two 7-bit, odd parity, ASCII characters. Timing is provided by a clock of the correct frequency which uses the run-in bits to phase the clock with the data. This is usually done with a phase-locked loop, or, in the case of the VBI explorer, with an LC self-syncing tuned circuit.

full time stamp sequence is sent only when the fractional seconds code is 2 or 3, otherwise just the framing code and fractional seconds are transmitted. The bits labeled XXXXXX vary from network to network and their purpose is unknown to me.

The time stamp used by many syndicated shows is the same as for network shows, but with an addition. During lines where the fractional seconds code is equal to 20, 22, and 24, the remainder of the bits form five 7-bit, no parity, ASCII upper-case characters. The contents of these 15 characters varies from show to show, but are usually grouped into five characters of syndicator identification, five of show name, and five of a unique show number. For example, a "Family Ties" show ID might be "PARTV FAMTI 106" (PARTV standing for Paragon TV]. Duplicate information is sent during lines with fractional seconds of 21, 23, and 25. The VBI Explorer's CLOCKSET command will display the time as well as this identification information if present.

VBI EXPLORER-OVERVIEW

A block diagram of the VBI Explorer is shown in Figure 3. Composite video is buffered and sent both to a video sync detector and a video clamp. The sync detector, built around the LM188 1 sync detector chip, generates composite sync, even/odd field signals, and vertical sync information. It also generates a burst signal at the beginning of each line that is used by a clamping circuit to clamp video to ground during the front porch so ground (0 volts) can be used as a logical zero for data detection. The sync detector's output is also fed to the CPU and used for VBI line counting.

Clamped video is then sent to a pair of *data slicers*. The fast data slicer, used in the WST, NABT, and time stamp circuits, has its slicing level set by the CPU via an S-bit DAC. A calibration mode lets the CPU sample data at various slicing levels from which it chooses the level with the lowest error rate. The slow data slicer, used for closed captions, employs a peak detector to determine

the optimal slicing level. Both data slicers generate TTL-level data as output.

TTL logic level output from the fast and slow data slicers is sent to one of two clock regeneration circuits which reconstructs the clock signal from the data. For time stamp detection, data from the fast data slicer triggers a 4.032-MHz clock divided by four to provide the necessary 1.008-MHz clock signal. Under CPU control, a 74HCT153 4-to-1 multiplexer selects the particular clock and data signal needed and routes them to the serial memory and clock circuits.

Everything comes together at the serial memory and counter block. Under CPU control, the software generates Valid-Line, which is set high during each VBI line to be read. When high, it causes a low-to-high transition of Data-Clock to store the current value of Data into a 4096x1-bit memory whose address is set by a 12-bit counter. The counter is incremented after each write. Once the VBI is over, Valid-Line is set low. This

C COMPILER FOR PIC CONTROLLERS

- Integrated software development environment including an editor with interactive error detection/correction.
- Access to all hardware features from C.
- Includes libraries for FIS232 serial I/O and precision delays.
- Efficient function invocation mechanism allowing call trees deeper than the hardware stack.
- Special built-in features such as bit variables optimized to take advantage of unique hardware capabilities.
- Easy to use high level constructs:

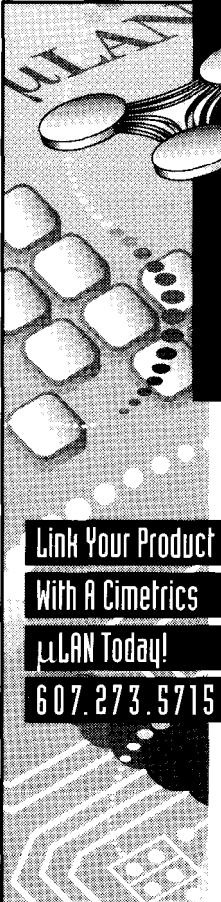
```
#include <PIC16C56.h>
#use Delay(Clock=2000000)
#use RS232(Baud=9600,Xmit=pin_1,RCV=pin_2)

main 0 {
    printf("Press any key to begin\n");
    getc();
    printf("1 khz signal activated\n");
    while (TRUE) {
        output_high(pin_8);
        delay_us(500);
        output_low(pin_8);
        delay_us(500);
    }
}
```

Special Introductory Prices

C5x compiler \$50	2 day shipping \$5
C71 compiler \$50	Next day shipping \$12
	COD additional \$5

CCS, PO Box 11191, Milwaukee WI 53211
414-781-2794 x30



Cimetrix
TECHNOLOGY

*Linking Microcontrollers.
Breaking Boundaries.*

The 9-Bit Solution

The Cimetrix Technology O-Bit Solution is a complete microcontroller network (µLAN) that supports the 8051, 68HC11, 80C186EB/EC, and many other popular processors. The Y-Bit Solution takes full advantage of microprocessor modes built in to microcontrollers. The 9-Bit Solution allows simple and inexpensive development of master/slave multidrop embedded controller networks.

- 8051, 68HC11, 80C186EB/EC compatible
- A full range of other processors supported
- Up to 250 nodes
- 16 Bit CRC error checking with sequence numbers
- Complete source code included

120 West State Street • Ithaca, New York 14850
Ph 607.273.5715 • Fx 607.273.5712

Link Your Product
With A Cimetrix
µLAN Today!
607.273.5715

gives the CPU control of the counter, which it resets and reads the stored data back at its own pace, converting the bits into ASCII characters. The combination of counter and memory operates as a very large shift register—storing the data during the VBI and spitting it back out again later. This scheme is necessary because the 8051 cannot keep up with the data rate of the closed caption text or network clock data.

Finally, the CPU is connected to both external data RAM and program ROM. The RAM is used primarily to store large amounts of data during high-speed teletext (WST, NABTS) operations.

HARDWARE-THE DETAILS

The core of the VBI Explorer (see Figure 4) is an 8031 microprocessor with 32K×8 EPROM (27256) and 32K×8 SRAM (43256). An 11.0592-MHz clock allows for standard baud rates. Serial I/O is converted to and from RS-232 levels by a MAX232 (U4).

An LF347 JFET op-amp (U6c) acts as a video buffer, feeding a LM1881 sync detector (U5) through a low-pass filter made up of R2 and C8. The LM1881 provides composite sync, burst (also known as front porch), vertical sync, and field outputs. The field output is especially helpful for network clock and closed caption data since these occur only during odd fields. The vertical sync output is inverted and presented to INT1 on the 8031. If software enabled, the 8031 will generate an interrupt on the trailing edge of each vertical sync pulse (the 8031 allows interrupts to be either edge or level triggered—in this project they are always edge triggered).

Another JFET op-amp (U6d) also acts as a video buffer, providing a signal which is clamped to ground

during the front porch of each line. The clamping circuit is driven by the burst output from the LM1881, but the burst output is too long and corresponds with the beginning of the clock run-in, so it must be shortened. This is done by a 74LS123 one-shot (U7a) which drives the clamping circuit (Q1,

from the DAC0800 (U22) is converted to voltage by an LM1458 op-amp (U21a). R21, R26, and R7 are set to provide an output voltage range of 0 to 0.5 volts. Since the actual slicing level is set by the error rate of the data, 5% resistors are more than adequate.

The outputs from the comparators

are open collector and are pulled up with 220- or 180-ohm resistors, resulting in an inverted TTL-level data stream. Inverters U8e and U8c are used to uninvert the bit stream since both upright and inverted data are needed for clock regeneration.

A pair of clock regeneration circuits, one for the closed-caption data and

the other for WST and NABTS, create a clock signal from the teletext data signal. We'll use the fast data clock regeneration circuit as an example—the slow clock regeneration circuit works on the same principle, but because the frequency is lower, some component values differ. The capacitor/resistor combination of C16/R18 acts as a differentiator, generating a spike upon each low-to-high transition of the data. The inverted data goes through a similar combination (C17/R19) causing a spike with each high-to-low transition. The outputs of both spikes are inverted by U11 and their open-collector outputs are ORed together so the output reaching U12c (pin 9) consists of a spike each time the data changes level. An 820-pf (C22) and a 560-pf (C18) capacitor across a 1.7–3.9- μ H variable inductor (L2) form a tuned circuit whose frequency can be adjusted to the bit rate of the data (5.72 MHz in this example). The data transition spikes cause the clock to be in phase with the incoming data, resulting in a stream of clock pulses used for data extraction.

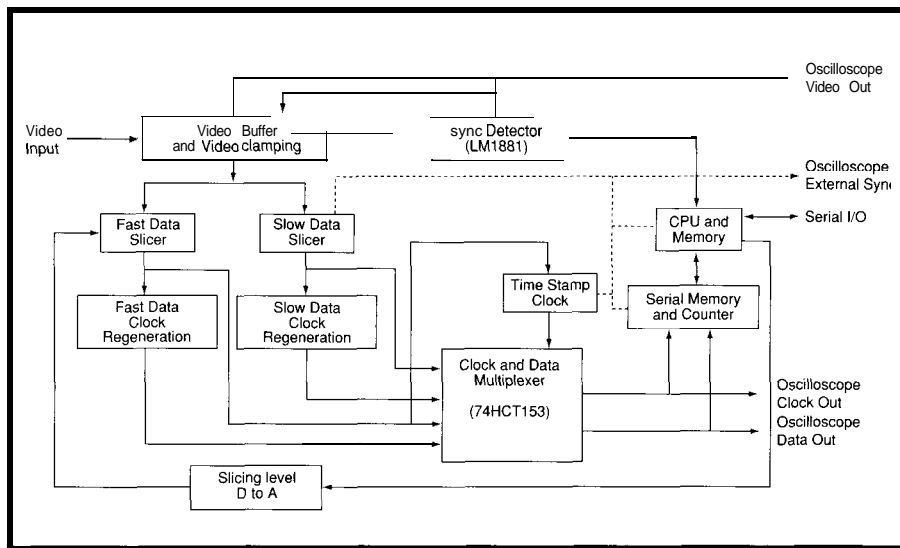


Figure 3—Much of the VBI explorer is involved with the clocking of the various signals found in the vertical blanking interval. If only specific signals are of interest, only portions of the circuit need to be built.

R4). After being clamped, the video is fed to an LM319 dual comparator (U10) for data slicing as well as the closed-caption peak detector.

Two JFET op-amps (U6a,b) form the heart of the closed-caption peak detector. The peak detector is allowed to charge up a 3300-pf capacitor (C13) whenever a valid line is being received. During other times, Q2 is turned on, shorting C13 to ground through R12 and resetting the detector. A lo-turn, 10k trim pot (R13) is used as a voltage divider and should normally be set near its midpoint.

Data slicing is done by an LM319 dual comparator (U10a,b)—one comparator each for the fast (WST, NABTS, time stamp) and slow (closed caption) data streams. The slicing voltage is fed to the noninverting input, while the clamped video gets the inverting input. Slicing voltage for the fast data slicer is supplied by an 8-bit DAC (U22). The CPU generates Slice-Data and Slice-Clk, which form the data and clock inputs, respectively, for an 74LS 164 8-bit serial-to-parallel shift register (U19). The output current

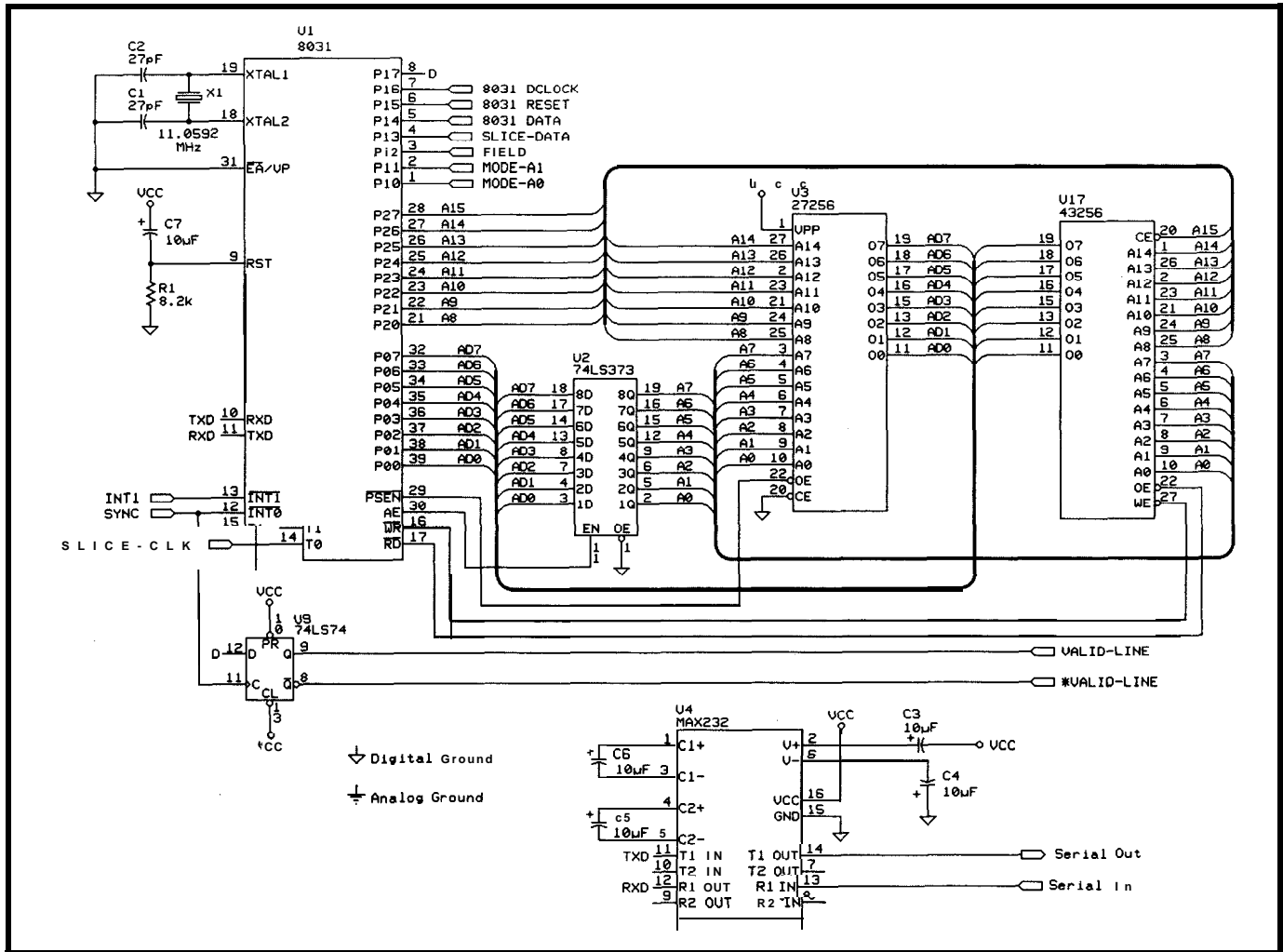


Figure 4a—The core of the VBI explorer is the same 8031 controller circuit typically used. A MAX232 provides RS-232 level shifting.

Since this system will only be working with one type of signal at a time (WST, NABTS, Closed Caption, or time stamp) there needs to be a way to take the clock and data information for each mode and select the one data and clock signal of interest. U13, a dual 4-to-1 line multiplexer, does the trick. The CPU (under software control) places a number from 0 to 3 on the multiplexer's address lines via I/O port pins P1.0 and P1.1. An address of "0" selects the closed-caption data and clock, "1" for WST and NABTS, "2" for the network time stamp, and "3" is not used. The outputs from the multiplexer consist of a single data signal and its corresponding clock signal which are used by the data recovery circuit.

Now let's review the sequence that occurs on each interrupt and see how the hardware and software interact to capture the data. To start

with, a table is set up in 8031 data memory consisting of an ordered list of VBI lines to be read, as well as which fields (even, odd, or both) that are needed. Assume all interrupts are turned off at this point. Suppose we are interested in closed-captioned data. Setting P1.0 and P1.1 to 0 causes only the closed-captioned data and clock signals to be passed by U13. Pulsing P1.5 briefly from high to low clears the 74HCT4040's 12-bit counter (U15) and sets the memory address to zero. P1.6 will be used to increase the counter when the computer is reading memory, so it is set low for now. P1.7 (which I'll discuss in a moment) is also set low.

Everything is ready, so interrupts are enabled on INT1, which will interrupt the processor at the end of the vertical sync pulse. The first thing to do upon the vertical sync interrupt is to store the address of the VBI table

as well as the current VBI line number minus one in registers. A short software delay loop is executed to get past the vertical sync equalizing pulses. Then the code enables horizontal interrupts to occur via INTO and exits the vertical sync interrupt routine.

The basic cycle time of an 1.1-MHz 8031 is between one and two microseconds, which is fast but not fast enough to have the computer read and store the bits as they come across. It's not even fast enough to have the computer interrupt at each horizontal sync pulse during the VBI and turn on the hardware to capture the data. By the time that would happen, bits would be lost. Instead, this system gets things started one line ahead of where the action is. When a horizontal sync interrupt occurs, it will increment the current line number and look at the first entry in the VBI line table. If it

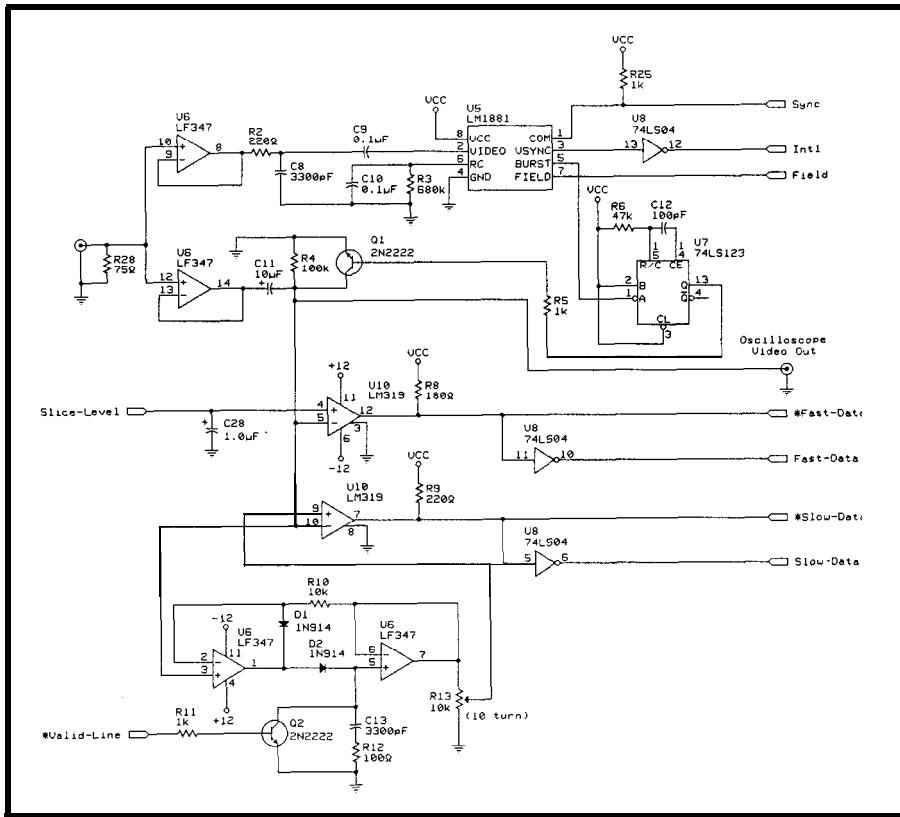


Figure 4b—An LF347 JFET op-amp acts as a video buffer, feeding an LM1881 sync detector through a low-pass filter made up of R2 and C8. The LM1881 provides composite sync, burst, vertical sync, and field outputs.

matches, P1.7 is set high. The line table pointer is then incremented to the next entry. If the system is receiving the end of the VBI lines [line 23], horizontal interrupts are turned off and a flag is set so the rest of the software can know that this VBI is done.

P1.7 is the data input of an 74LS74 D flip-flop (U9b). It is set high by the line that arrives *ahead* of a teletext line. The clock input is the composite sync signal, so the output from the flip-flop goes high at the beginning of the next line. It is reset by setting P 1.7 low during the next horizontal sync interrupt, which clocks a 0 at the start of the next line (unless it is set high by the software to capture the next line also). The Q output from this flip-flop is labeled Valid-Line and is high during each VBI line we are capturing. It serves to allow the board to switch between capturing data and allowing the CPU to read it back.

Now all that's left is to clock the data into memory and advance the memory address. Another 74HCT153 dual 4-to-1 multiplexer (U14) is configured as a folded 3-input truth

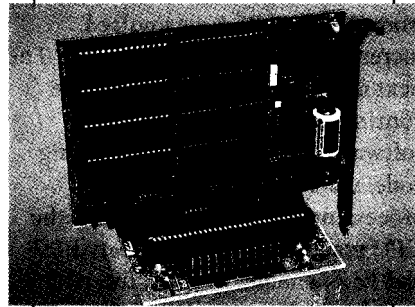
table. When Valid-Line is high, the data clock is allowed to pass through and generates a signal to fire an 74LS123 one-shot (U7b), creating a memory-write pulse. When Valid-Line is low, it lets the output from P1.6 act, as the clock instead and does not generate a write pulse.

The memory circuit consists of a 12-bit counter (U15) generating the address for a 2147 4096x1-bit SRAM (U16). If you were building a version of this circuit for closed captioned or network time stamp use only, you could get away with a 1024x1 (or smaller) memory.

When Valid-Line is high and the system is writing data to memory, the memory address is advanced one count for each data clock pulse. If there are multiple VBI lines to be scanned, data from new lines is added after the previous data. There may be a few bits of garbage generated in between lines, but the framing codes will let the system bypass that.

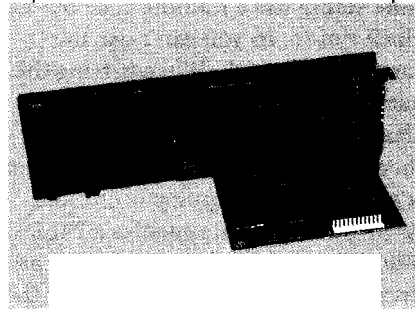
Once the VBI lines have been read and the horizontal interrupts turned off, the software detects a flag set during the last horizontal interrupt.

VMAX
QUALITY PRODUCTS
RESPONSIVE SERVICE
RELIABLE DELIVERY



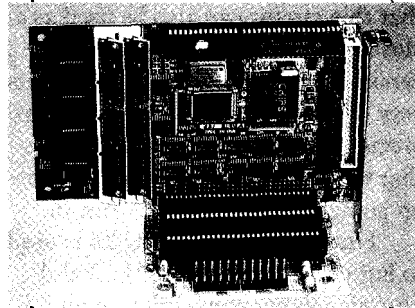
SOLID STATE DISK — \$135"

1/2 Card 2 Disk Emulator
 EPROM, FLASH and/or SRAM
 Program/Erase FLASH On-Board
 1M Total, Either Drive Bootable



25MHZ 386DX CPU — \$695"

Compact AT/Bus or Stand Alone
 On-Board SVGA, IDE, FDC, 2 Ser/Bi-Par
 FLASH/SRAM Drives to 2.5M
 Cache to 128K, DRAM to 48M



**TURBO XT
 w/FLASH DISK — \$266***

To 2 FLASH Drives, 1M Total
 DRAM to 2 M
 Pgm/Erase FLASH On-Board
 CMOS Surface Mount, 4.2" x 6.7"
 2 Ser/1 Par, Watchdog Timer

All Tempustech VMAX products are
 PC Bus Compatible. Made in the
 U.S.A., 30 Day Money Back Guarantee
 *QTY 1, Qty breaks start at 5 pieces,

TEMPUSTECH, INC.

TEL: (800) 634-0701

FAX: (813) 643-4981

Fax for 295 Airport Road
 fast response! Naples, FL 33942

This tells the software, "it's my turn now," and the data crunching begins. P1.6 gets pulsed, which resets the 12-bit counter so the current memory address is 000. The first data bit is read from P1.4 and P1.6 gets pulsed, incrementing the counter to 001. The next data bit is read and the cycle continues. In the meantime, the software is looking for the framing code, reading the data bits, and processing the data. Once done, the CPU resets the counter, clears P1.7 and P1.6, enables vertical interrupts, and the cycle restarts.

One last thing. The circuitry to generate the network time stamp data is a little different from that used for teletext. This is because the network time stamp has a framing code but no clock run-in, so you can't use the tuned circuit trick. When getting the network clock, a D flip-flop (U9a) is cleared until Valid-Line goes high. It then waits for the first data bit. When it goes high, the flip-flop is clocked, setting Q high. This enables a 74LS93 4-bit ripple counter to start acting as a divide-by-four counter, generating the necessary clock signal. The flip-flop is reset when Valid-Line goes low. It is interesting to note that the clock speeds for the network time stamp and closed-captioned data are close enough that this circuit can also be used as a closed-caption clock—indeed, early versions of the VBI Explorer used this very scheme. It requires, however, a precise slicing level and a guarantee that the first data on the line be the up-going pulse of the clock run-in.

Finally, a 74HCT04 (U20) is used to buffer clock, data, and the Valid-Line signals for connection to an oscilloscope. A NE5090 relay driver (U24) is used to drive channel changing relays for automatic control of a VCR. Hopefully your VCR has an up-and-down channel switch instead of a separate switch for each channel. You'll need to modify this section, as well as a VCR, if you want automatic channel switching.

SOFTWARE

User interaction with the VBI Explorer is through a set of about 50 commands, many of which are listed

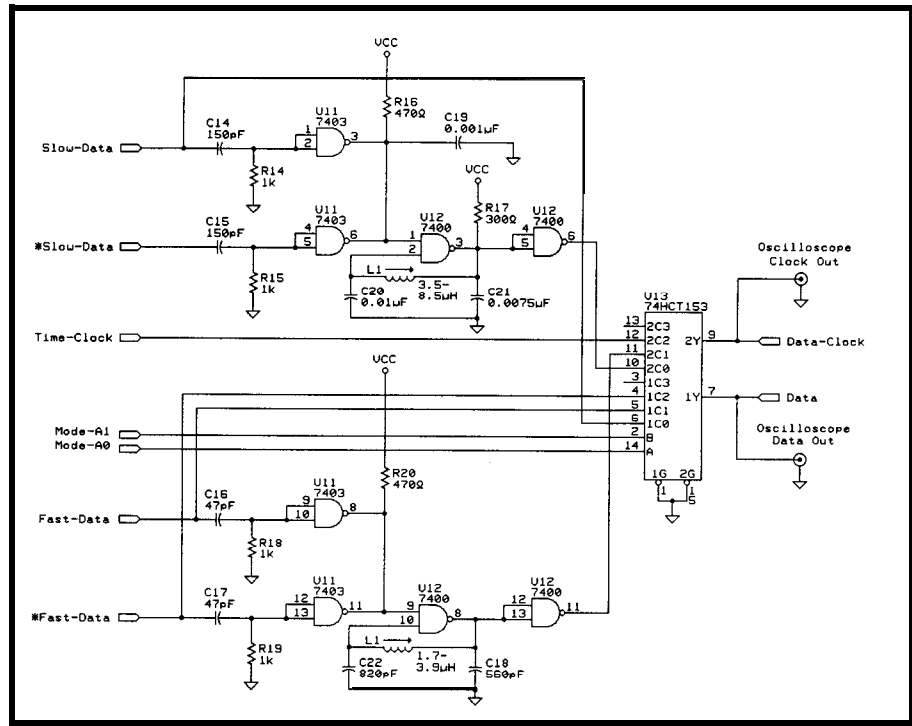


Figure 4c—A74HCT153 multiplexer is used to select one of four modes: closed captioning, WST, NABTS, or time stamp.

in Table 2. On-line help is available for all commands.

Each type of teletext has its own group of related commands. Closed caption and time stamp commands automatically scan lines 21 and 20, respectively, of field 1, but this can be disabled to read caption or clock data from any VBI line or field. WST and NABTS require that you manually specify which lines and fields to use. Usually this is achieved by combining the BOTH command with a sequence of numerical line commands. For example, to read WST data from the even and odd fields of lines 14 and 15, use the BOTH, 14, and 15 commands in sequence. A number of commands which set commonly used WST and NABTS line sequences are provided.

Setting the correct slicing level for everything other than closed captions is done through a series of calibration commands, one each for the WST, NABTS, and time stamp signals. Calibration works by scanning through a range of slicing levels and keeping a table of errors. A weighted average then calculates the level with the lowest error rate. The process is automatically repeated until the error rate becomes stable, at which point that value becomes the new slicing

level. Optimum levels tend to stay around the same point, especially for network time stamp signals, therefore a set of default levels are provided which can be set in ROM once the optimum levels for your video source are known. You can also set the slicing level manually.

CONSTRUCTION

The current VBI Explorer is built using wire-wrap techniques on a ground plane perf board. Because of the potential noise problems that can occur when computers and video signals are in close proximity, I strongly recommend using a board with a ground plane. Keep analog and digital components separated as much as possible and use a separate analog ground connected to the primary ground only at one point. The RC circuit (R3, C10) that forms the timing for the sync detector is particularly critical, so use good-quality components that are heat stable (such as a metal film resistor for R3). Best stability is obtained when non-LS TTL parts are used for U11 and U12. Likewise, use 74HCT parts for U13 and U14. Keep all component leads as short as possible. Bypass all chips, including the analog ones, with 0.1- μ F

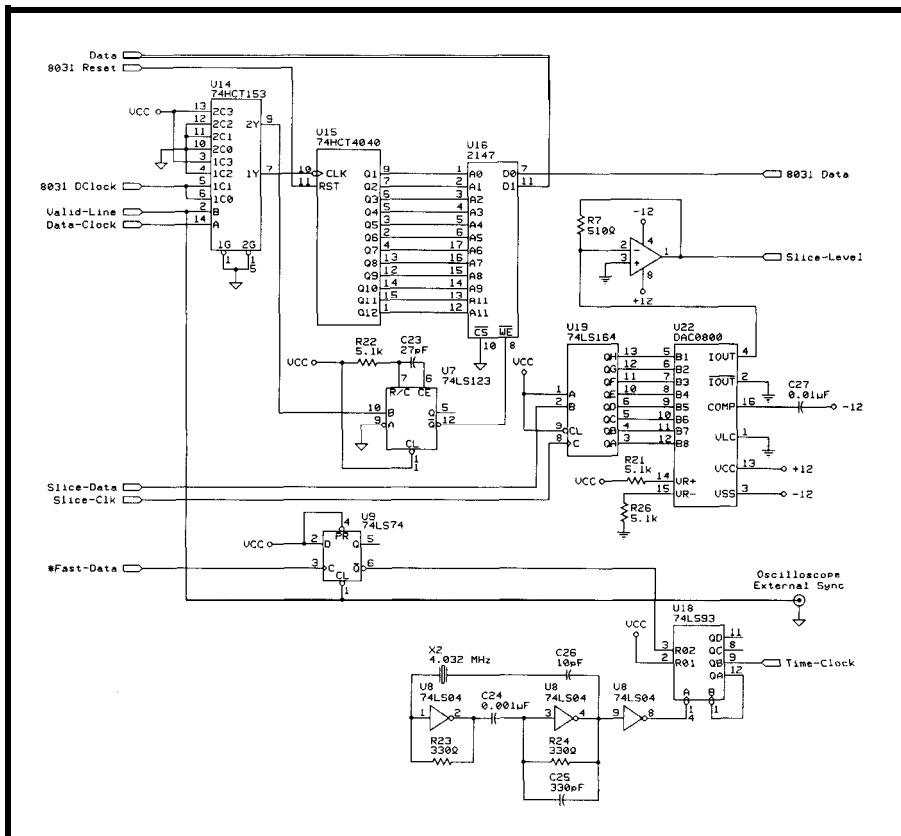


Figure 4d—The slicing level can be set dynamically by the firmware to minimize errors. Once the processor decides to capture data from a particular line, data is automatically captured and stored in a 4096x1 RAM chip (U6).

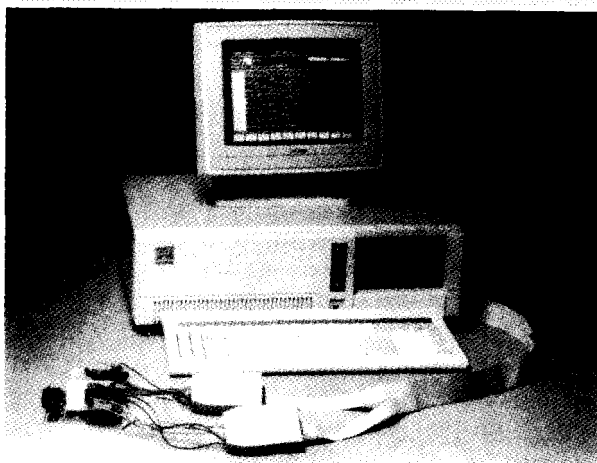
disc capacitors. Since capacitors are essential to many of the timing circuits, use high-quality components, such as polystyrene, for C8–10, C12–13, C18–19, and C20–25. Except for R3, ¼-watt 5% carbon resistors are sufficient (if ventilation is adequate, a 5% carbon resistor can be used for R3). C28, a 1.0-μF electrolytic capacitor, used to filter digital hash from the slicing level should be mounted close to pin 4 of U10a. Use an RF-tight enclosure to prevent interference with the video signal.

TUNE-UP

Once you have constructed the Explorer and done preliminary voltage tests, connect a 9600-bps terminal and apply power. If the memory and CPTJ circuits are operational, you should get a greeting on the screen. Try typing the HELP command to ensure that characters can move in both directions.

Now apply a standard 1-volt, 75-ohm composite video signal to the video input and view the FIELD,

PC-Based Logic Analyzers



Sophistication at Low Cost

ID160 (50MHz) \$595

ID161 (100 MHz) \$695

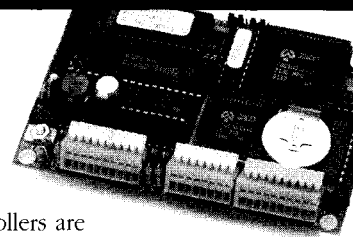
- *High Speed • 8K Trace Buffer • 16 timing channels expandable to 32 state channels
 - *Multi-Level Triggering
 - *State Pass Counting
 - *Event Timer/Counter
 - *Performance Histograms
 - *Hardcopy Output
 - *Disassembles 8-bit micros
 - *Supports VGA /EGA/HGA
 - *Demo diskette available
- 30 Day Money Back Guarantee



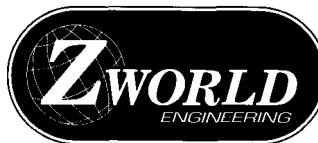
INNOTECH DESIGN, INC.
6910 Oslo Circle, Suite 207
Buena Park, CA 90621
Tel: 714-522-1469 FAX: 714-527-1812

LOW COST MINIATURE CONTROLLERS

Save time.
Save money.



Our wide range of C-programmable miniature controllers are ideal for control applications, data acquisition, and test and measurement. Compact and low in price (the Little PLC™ above is 2"x3" and \$195), these controllers are programmed with our easy-to-use Dynamic C™ development system. Our controllers feature digital I/O, ADCs and DACs, relays and solenoid drivers, RS232/RS485 serial ports, battery-backed memory and time/date clock, LCDs, keypads, enclosures and more!



1724 Picasso Ave.
Davis, CA 95616
916.757.3737
916.753.5141 FAX

24-Hour AutoFAX 916.753.0618. Call from your FAX. Request catalog #18.

VSYNC, and COM pins on the LM1881 with a scope to ensure proper sync detection. Now look at the Oscilloscope Video Out signal. If a video signal is present, the video clamp is working properly. For a low error rate, your video signal must be virtually noise free. Small amounts of noise can be tolerated by closed captions and the network time stamp, but not with the faster forms of teletext.

First, get the network time stamp operational. Feed a video signal from ABC, CBS, or NBC during prime time into the Explorer. Type C LOCK and observe the signal from Oscilloscope External Sync. If the system is working, you should see a single pulse occurring every 1/30 of a second. Connect this line to your scope's external sync such that it triggers on an up-going pulse. Hook the Oscilloscope Video Out to the scope and you should be viewing the network time stamp—little pulsating blips with a big blip every second. Type CCAL to obtain an optimum slicing level. The calibration command takes anywhere from 10 to 40 seconds to work. Now type C LOCK again; at this point the CRT should be displaying the time. Typing any character exits you from the clock display back to the monitor. Tune to other network stations and repeat the process. Find a slicing value which works across all three networks, change the default value in the source to this value, then reassemble. The C LOCK command should then work without having to recalibrate.

To set up for closed-caption display, tune to any program that has captions. Type CCC and observe the video signal. You should see a raw closed-captioning signal that resembles a rounded version of Figure 1. Adjust R13 to near its midpoint and look at the Data Out signal. Adjust it further so that the clock run-in resembles a square wave with a 50% duty cycle (Figure 1). Now adjust L1 until the CRT output resembles strings of ASCII interspersed with command codes. Tune L1 for best results. If you have a frequency counter, connect it to U12b pin 6, disconnect the video, and adjust L1 to

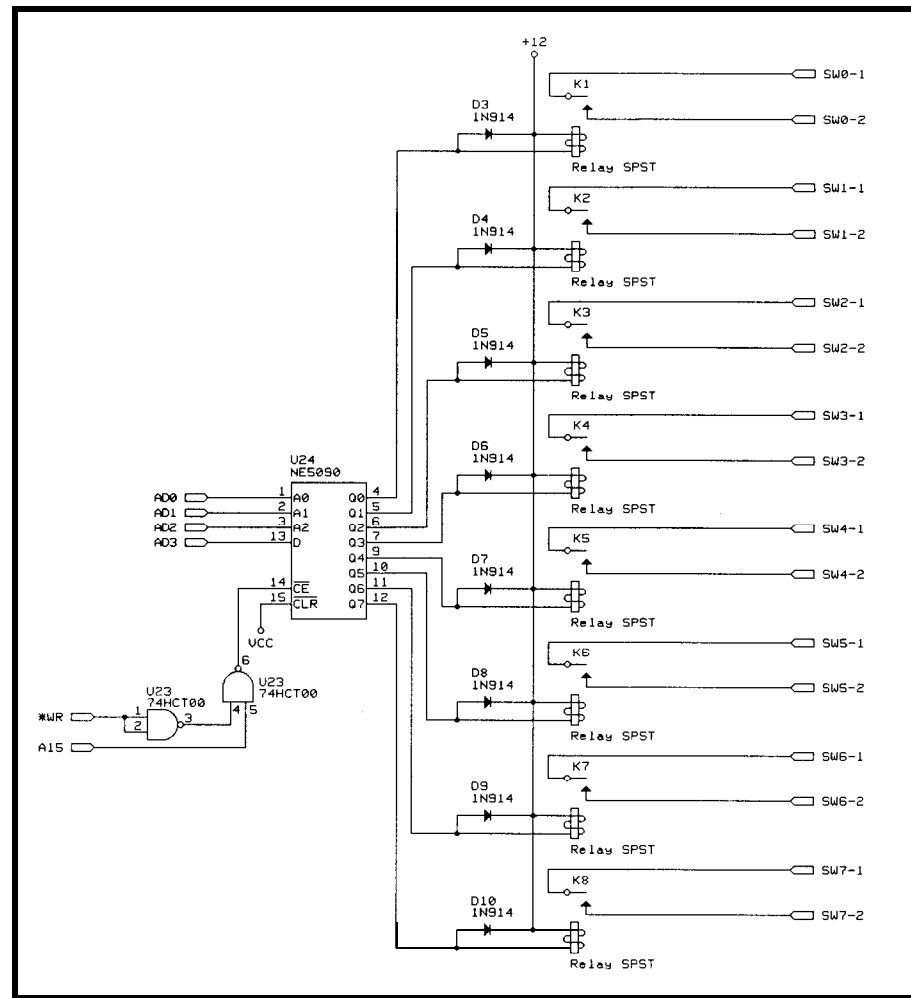


Figure 4e—An NE5090 relay driver is used to drive channel-changing relays for automatic control of a VCR.

a clock rate near 1.007 MHz. Once L1 is properly set, it should be left alone. Tune to other networks and stations which have closed-captioned signals and adjust R13 so that the clock run-in from Data Out best resembles a square wave for all stations. Type a space to get out of command mode and type CC. A decoded version of the closed captions will appear.

The last features to set up are the fast teletext signals: WST and NABTS. Unfortunately, this will be largely theoretical since both WTBS and CBS no longer carry teletext magazines as of 1994. Let's assume that a WST signal appears on VBI lines 17 and 18. Type SETWSTH (a shorthand for setting capture of lines 17 and 18) followed by WSTPARITY. Look at the Video Out signal and assure yourself that VBI lines 17 and 18 have teletext on them. Type a space to get back to the monitor and type W U S E 6 8. This tells the Explorer to use a value of 68 hex

for the slicing level. Type W ST PA R I TY and observe the clock run-in signal via Data Out. If it is not a square wave, try different values for the W U S E command (values can range from 00 to FF). Once you have a square wave, type W ST PA R I TY again and tune L2 for the lowest number of parity errors. If you have a frequency counter, connect it to U12d pin 11 and adjust for 5.7272 MHz after disconnecting the video. Once you have minimized the number of parity errors, type WC A L to find the optimum slicing level. Use W ST PA R I TY again to fine tune L2 for minimum errors. The parity error count should stay on 00 except for an occasional error. Some scope probes load the circuit, so if you can't get 00 errors, remove the probe and try again using the parity error display to guide you. Again, once L2 is properly set, you shouldn't mess with it. Video levels will switch slightly from time to time necessitating reuse of the W CA L

Miscellaneous Commands		wstbuffer	Display buffer's worth of data in printable format
help <cmd>	Prints help for given command	wstscan	Read VBI lines and immediately display
12-22	Set VBI lines to scan	wstraw	Send raw WST data directly to serial port
setwst	Shorthand for 15, 16, 17, 18, both	wstclean	WST commands will filter out graphic characters
setcbs	Shorthand for 15, 16, 17, both	wstall	WST commands will send graphic characters
setwstl	Shorthand for 15, 16, both	wstparity	Read VBI lines and display parity errors
setwsth	Shorthand for 17, 18, both	wcalibrate	Calibrate slicing level for WST
scope	View VBI lines on oscilloscope	wuse <nn>	Set WST slicing level to <nn> (00-FF)
clear	Clear VBI line settings	NABTS Commands	
odd	Interrupt on odd fields only	nabt	Collect buffer's worth of NABT/NAPLPS data, decode into pages, and selectively display
even	Interrupt on even fields only	nahex	Like "nabt" but display NAPLPS code in hex
both	Interrupt on both odd and even fields	nabinary	Like "nahex" but sends raw data for further processing
terminal	Use cursor control for displays	nabuffer	Display buffer's worth of data in printable format
show	Display current line, field, and slicing values	naparity	Read VBI lines and display parity errors
default	Use lines 21 and 20 for closed captions and clock commands (respectively)	nascan	Read VBI lines and immediately display
nodefault	User-specified VBI lines are used for closed captions and clock commands	naraw	Send raw NABTS data directly to serial port
Closed Caption Commands		ncalibrate	Calibrate slicing level for NABTS
cc	Decode and display closed captions	nuse <nn>	Set NABTS slicing level to <nn> (00-FF)
cctext	Decode and display text channel	Network Time Stamp Commands	
ccclean	Display raw captions with control characters converted to "(char)"	clock	Read time from network and display with updates
ccraw	Output raw captions without any processing	clockset	Same as clock, but only displays time once, and displays program id information if present
ccnofill	Same as "cc" but no line filling is done	clockraw	Display raw time code in binary
World Standard Teletext Commands		ccalibrate	Calibrate slicing level for network time stamp
wst	Collect buffer's worth of data, decode into pages and selectively display.	cuse <nn>	Set clock slicing level to <nn> (00-FF)
pg <nnn>	Scan for page <nnn> and display.		

Table 2—The VBI explorer supports four kinds of teletext, each with its own set of commands. There are also several general-purpose commands used during setup.

command. Now type PG 100. Within a few seconds the index page should appear. Notice that once the page is displayed, you are not placed back in monitor mode. This is so if the page is updated, or has subpages, they will automatically be displayed as they change. Type a space to return to the monitor.

For NABTS, we will assume three lines of VBI data are present on lines 15, 16, and 17. Type SETCBS (shorthand for capturing lines 15–17). Observe Video Out and confirm three VBI lines of teletext. If all looks correct, type NC A L followed by N A P A R I T Y and confirm that the error rate is low. Type N A B U F F E R and wait several seconds. A printable version of the raw NATBS code will appear. Type a space to stop the output and return to the monitor. Assuming a low error rate, type NABT and wait for it to print a page number and ask if you wish it displayed. Typing Y will print a reasonable facsimile of the textual portions of the page mapped to a 24-line by 41-character screen. As usual, a space will return you to the monitor.

NETWORK DIFFICULTIES

Problems with reception may not be due to a problem with the decoder. More than once have I wasted countless hours trying to fix a bug only to discover the problem originated with the network. When experiencing difficulties, always look at the signal on the scope to make sure it is both present and reasonable. Attenuation of the clock run-in signal is a common occurrence with both WST and NABTS. Diagnosis is obvious by looking at the scope, but painful otherwise. Occasionally, the signal will disappear or selected VBI lines will be shuffled. Closed captions may appear on lines other than 21.

Remember that captions and other services are copyrighted and that copyright laws apply to their use. Also, services have been known to change which VBI lines they use from time to time. There are an increasing number of proprietary VBI formats found on cable TV, so don't be disappointed if you can't copy everything.

If you use a VCR for a video source, be aware that older VCRs may

have poor automatic gain control (AGC), causing large voltage changes when the picture shifts from mostly white to mostly black or vice versa. This causes a higher error rate for teletext and network clock signals. Newer VCRs (8 years old or less) tend to be more reliable. □

The author would like to thank *Jose Sancho and Karen Barnes for their help in preparing this article.*

Mike Barnes holds a B.S. in Computer Science from the University of North Texas and an M.D. from the University of Texas Health Science Center at San Antonio. He is a Family Practitioner with an interest in Medical Informatics.

SOFTWARE

Software for this article is available from the Circuit Cellar BBS and on Software On Disk for this issue. Please see the end of "ConnecTime" in this issue for downloading and ordering information,

REFERENCES

ANSI Standard X3.110-1983: Video/Teletext Presentation Level Protocol Syntax (NAPLPS), December 1983. The official NAPLPS standard. Essential for decoding NAPLPS, but Fleming's series in *Byte* is almost as complete.

Benzel, J., and Nuckolls, L.R., "Closed Captioning with the Motorola 68HC05CC1," *The Computer Applications Journal*, May 1993, pp. 12-24. Good introduction to Closed Captioning.

Brand, S., *The Media Lab: Inventing the Future at M.I.T.*, Penguin Books, New York, 1988. Excellent introduction to the idea of personalized information systems.

Chambers J.P., "Enhanced UK Teletext Moves Towards Still Pictures," *IEEE Transactions on Consumer Electronics*, Vol CE-26,

August 1980. pp. 527-554. Entire UK Teletext Specification is listed as an appendix.

Crowther, G.O., "Adaptation of U.K. Teletext System for 525/60 Operation," *IEEE Transactions on Consumer Electronics*, Vol CE-26, August 1980. pp. 587-598. Explains the use of the "gearing" line.

Electronics Industries Association (EIA), Joint EIA/CVCC Recommended Practice for Teletext: North American Basic Teletext Specification (NABTS). EIA-516. May, 1988. The official NABTS standard.

Electronics Industries Association (EIA), Line 21 Data Services for NTSC. EIA-608, 1992. The official Closed Captioning standard.

Fleming, J., Frezza W., "NAPLPS: A New Standard for Text and Graphics:

Part 1:" *Byte*, Feb. 1983, pp. 203-254. Part 2: *Byte*, March 1983, pp. 152-185. Part 3: *Byte*, April 1983, pp. 190-206. Part 4: *Byte*, May 1983, pp. 272-284. Excellent detailed introduction to NAPLPS, almost a reference in itself.

Holland, G.L., "NAPLPS Standard Defines Graphics and Text Communications," *EDN*, Jan. 10, 1985, pp. 179-192. Nice overview of NAPLPS. Has examples of disassembled NAPLPS code.

Mothersole, Peter L., and White, Norman W., *Broadcast Data Systems: Teletext and RDS*. Butterworths, 1990. Detailed description of the WST system.

IRS

404 Very Useful
405 Moderately Useful
406 Not Useful

STOP! **LOOK!** **LISTEN!**

STOP! Odds are that some time during the day you will stop for a traffic signal, look at a message display or listen to a recorded announcement controlled by a Micromint RTC180. We've shipped thousands of RTC180s to OEMs. Check out why they chose the RTC180 by calling us for a data sheet and price list now.

MICROMINT, INC.
4 Park Street, Vernon, CT 06066
(203) 871-6170 • Fax (203) 872-2204

CALL 1-800-635-3355

Employer Ownership of Employee and Consultant Work Product

FEATURE ARTICLE

Mary M. Luria
& Laura Butzel

a

basic rule of copyright law, known as the "work for hire doctrine," is

that an employer automatically and without any written employment or other contract owns copyright in the works of employees which are prepared within the scope of employment.¹

This rule applies to a wide range of employee works covered by the copyright law: writings, art, photography, music, computer programs, and other works. While this would appear to be a straightforward rule, over the years a number of different interpretations of what constitutes an employee or employment for purposes of this doctrine arose. In *Community for Creative Nonviolence v. Reid*,² the Supreme Court attempted to define who is an employee for purposes of the doctrine and to put an end to the ongoing conflict among the circuit courts. Despite the Court's decision in *Reid*, it has become clear that many uncertainties remain.

It is impossible to overestimate the importance of reducing to writing the rights of employees, employers, and independent contractors in creative work products. Any other approach leaves at risk the employer whose employees perform some of their creative work outside the traditional workplace, as well as the employer who commissions free-lance consultants. The risk is simple. The person who pays may not own copyright in the creative output of the other person who was paid to create.

Since, in the epic words of an earlier decision, "no one sells or

mortgages all the products of his brain to his employer by the mere fact of employment,"³ a line must be drawn between works prepared within the scope of employment, or employment-related works, and all other works—the latter of which is owned by the employee. The legal rules for this line-drawing are far from clear. The line is not subject-matter oriented: an employee can create computer programs for his employer and computer programs as a "hobby" or "in his spare time." Many, if not most, of the latter will be owned by the employee. The line is also not merely "time-of-day" oriented. The employer will own works created outside regular working hours if they clearly fall within the employee's duties and regardless of whether they were produced on or off the business premises or during or after work hours.⁴ Conversely, while the use of the work time, premises, and assets of an employer points strongly in the direction that the resulting work will be owned by the employer, this too is not totally dispositive, even for employees in jobs with expansive hours and 'round-the-clock job demands.⁵

This state of the law on works for hire presents particular problems in the case of employees who work off the premises (often at home) in areas of interest to the employer but not within the job description or pursuant to the employer's instructions. The number of computer programmers who work at home on computer-related hobbies makes this problem particularly acute in the software industry, although companies employing writers and visual artists have been

Do you have work-related hobbies or work at home, or are you an independent contractor hired on a part-time or project basis? If so, who owns your work? Find out the latest case law from two attorneys.

experiencing the problem for many years. Several courts have recently tried to deal with the problems affecting software, but with diverse results which leave the employer with some uncertainty regarding ownership.

COPYRIGHT OWNERSHIP

In *Avtec Systems Inc. v. Peiffer*,⁶ the District Court of the Eastern District of Virginia held that the employer did not own the copyright in such "spare time" computer software, since it was prepared outside working hours and space, not intended for the employer, and not done to fulfill work requirements, even though the software was related to the employer's business and the employee allowed the employer to use prior versions of the program.' In *Avtec Systems*, the employer was able to prevail on a separate trade secret misappropriation claim that the program in question incorporated its confidential proprietary information and that the employee had breached his fiduciary duty to the employer, with the result that a constructive trust was imposed on the benefits and revenues of the program; that is, Avtec received a nonexclusive right to use the program itself and a share of license revenues, but not the copyright in the program, or an injunction against its reproduction, distribution, or use.⁸

The message of the Avtec case for the employer is the employer, absent trade secret or similar claims, will not own works of the employee unless:

- a. they are of the type the employee was hired to create;
- b. the work is done substantially within the authorized time and space limits of the work environment; and
- c. the work is performed, at least in part, with the purpose to serve the employer.

On significantly similar facts, the District Court of South Carolina in *Miller v. CP Chemicals Inc.*⁹ held that computer programs developed by an employee, not hired primarily as a computer programmer, written at his home, on his own time, without

overtime pay (despite the fact that he was paid by the hour), and without the assistance of other employees or the use of company assets, belonged to his employer. The court, looking to the law of agency,¹⁰ found this work to be "incidental" to the type of work the employee was hired to perform as supervisor of the quality control laboratory and thus within the scope of employment. The court also found that the work was actuated "at least in part" to serve the employer." These factors were held to outweigh the time-of-day/workplace factors which the court conceded as "weighing heavily" in the employee's favor. The court summarized as its rationale:

"However, the driving force behind the creation of the computer programs was to benefit CP by making the quality control laboratory more efficient. Furthermore, the development of the computer programs was clearly incidental to the other work performed by Miller."¹²

In other words, the fact that the programs were created to simplify Miller's job and to eliminate errors, coupled with the fact that the programs dealt specifically with certain of the employer's products, resulted in an employer-owned work. Nonetheless, the facts of the case were equally consistent with the employee's argument that he owned the programs and licensed them to his employer only for the duration of his employment. Since the latter license arrangement was only set forth in a letter from the employee and was not signed by the employer (although it was generally known in the company and posted on terminals), it was ineffective to rebut the presumption of employer ownership, which arose once the court found the work to be created within the scope of the plaintiff's employment.

GET IT IN WRITING

The message of the *CP Chemicals* case for both employers and employees is clear: in order to vary the copyright law's presumptions about ownership,

there must be a written contract signed by both the employer and employee or, at a minimum, by the party whose presumptive rights would be restricted by the agreement.¹³ Such a contract can either broaden or circumscribe the rights of the employer—with the opposite result in regard to the rights of the employee. If it narrows the rights of the employer, it must be signed by the employer.

The employer may well want to avoid the uncertainties which result in cases like the two discussed here, to say nothing of the expense of litigation, by means of a written contract with the employee if there is a substantial danger that the employee will create what are arguably out-of-scope works of interest to the employer—usually described as works with significant application within the employer's basic businesses. Such a contract must be clear and specific; it should not speak vaguely of the employer's right to work produced by the employee at all times or to an employee's entire work product. It may provide that the employer has only a right of first negotiation or first refusal of such works and has to pay money beyond basic salary in order to secure the copyright or license rights in such works. The alternative, however, may be that these works are never seen until they are in the hands of third parties. This is an alternative many employers find unacceptable.

Most employers do not have written contracts with any of their employees or have such contracts only with a limited group of management and technical employees. Other employers have employee policies which are circulated to a wider group or to all employees. While such policies could and should address intellectual property ownership issues, they may not be adequate as a legal matter. In this area, the copyright law requires a bilateral signed contract or at least a contract signed by the employee (in a situation which restricts the employee's rights) or by the employer (in a situation which restricts the employer's rights) to vary the presumptions created by the law about scope of employment which are

discussed in the *Avtec Systems* and *CP Chemicals* cases. An unsigned policy statement may constitute some evidence of what the employer considered to be the employee's scope of employment, but it will not serve to transfer rights in an out-of-scope work owned by the employee to the employer; it might, at most, transfer to the employer a nonexclusive license, the only copyright transfer which does not require an agreement in writing, but rather only an understanding or agreement.

In the absence of an agreement, the uncertainties in determining whether a work is owned by an employer or employee can be great. Even more uncertainties arise when it is not clear whether a hired individual is an employee or an independent consultant. Recent case law strongly suggests that the employer may pay for, but will not own, copyright in works of individuals who work for pay on or off the employer's premises but do not receive benefits, unless there is a written contract signed by both in which work-for-hire status or employer ownership is expressly agreed.

INDEPENDENT CONTRACTORS

In *Aymes v. Bonelli*,¹⁴ the 2d U.S. Circuit Court of Appeals held that the most important features of the hiring relationship which distinguish an employee from an independent contractor are the putative employer's treatment of the employee for benefits and tax purposes. In this case, Aymes was hired to work as a computer programmer to create a series of programs over a period of two years. He never received health or insurance benefits. His employer never paid the employer's percentage of payroll taxes, nor did the employer withhold federal or state taxes on the employee's income, paying Aymes instead on a gross basis and providing a Form 1099, rather than a W-2 Form. This proved fatal to the employer's later claim of ownership of copyright in the programs written by Aymes. According to the 2d Circuit:

"The failure of Island to extend Aymes any employment benefits

or to pay any of his payroll taxes is highly indicative that Aymes was considered an outside independent contractor by Island. Indeed, these two factors constitute virtual admissions of Aymes's status by Bonelli himself. Moreover, they also point out a basic inequity in Aymes's treatment. Island benefited from treating Aymes like an independent contractor when it came to providing benefits and paying a percentage of his payroll taxes. Island should not in one context be able to claim that Aymes was an independent contractor and ten years later deny him that status to avoid a copyright infringement suit."¹⁵

Aymes had sued for copyright infringement after he left this job, based on his former employer's alleged breach of an oral agreement limiting the use of the computer programs in question to a single computer at a single site. The employer defended by claiming that, as a result of the employment relationship, it owned the copyright on the programs it was alleged to be infringing. The 2d Circuit considered as relevant a few other indicia of whether Aymes was an employee (employer's right to control manner and means of work, level of skill of work required, employer's right to assign additional work),¹⁶ but found the benefits/tax point dispositive.

The Aymes case was remanded to the trial court to consider whether the employer contributed enough to the creation of the programs so that it was a joint copyright owner and, therefore, not an infringer."

The message of the case for 2d Circuit (New York, Connecticut, and Vermont) employers is clear: if a hired individual is not an employee for benefits/tax purposes, he or she is unlikely to be treated as an employee for copyright purposes and, therefore, a written work-for-hire contract signed by both parties is necessary if the employer is to own the copyright on the "employee's" creative output. (The employer could also seek to receive in writing an assignment of the copyright or an exclusive license signed by the

employee.) According to the 2d Circuit:

"The importance of these two factors is underscored by the fact that every case since *Reid* that has applied the test has found the hired party to be an independent contractor where the hiring party failed to extend benefits or pay social security taxes."¹⁸

California takes the same position as a matter of statutory law, so several of the most commercially significant states are in agreement on this point.

There are a number of post-1989 cases in other geographic areas which have the same effect, but articulate the rationale less clearly." Many of these cases focus strongly on the issue of employer control and "employee" independence, rather than benefits/tax issues, and it now seems clear that these three issues will emerge as the most significant from the *Reid* case's longer list in distinguishing employees from independent contractors.²⁰

It should be added that in *Playboy Enterprises Inc. v. Dumas*,²¹ the District Court in the Southern District of New York stated that "work made for hire agreements must precede the creation of the work." The court found that works prepared by an independent contractor and artist were not specially ordered or commissioned and thus not works made for hire. However, the court stated that assuming the works were specially ordered or commissioned, the writings establishing the works as works made for hire did not precede the works themselves.²² As a result the artist owned the copyright in his work product.

FINALLY...

All too many employers use no contractual paperwork in arrangements with consultants, so that the only "documents" relating to the employment relationship or copyright ownership in the commissioned works are the checks and the Form 1099 issued to the consultant. Under these circumstances, the "employer" will be working with an independent consultant and the latter will own the

copyright, leaving the former with (at most) a nonexclusive license no matter how much development money has been paid. This will continue to be an important issue as long as the Copyright Act incorporates the work for hire doctrine.

There is some prospect that the portion of the doctrine which deals with independent contractors may be amended after decades of debate, but it seems less likely that the treatment of true employees will be changed. Even if the Copyright Act is amended, there will be an important group of post-1978 works which continue to be subject to the doctrine for the full term of copyright. □

Mary M. Luria is a partner of Patterson, Belknap, Webb & Tyler's Intellectual Property Group. Laura Butzel is associated with the firm.

IRS

- 407 Very Useful
- 408 Moderately Useful
- 409 Not Useful

REFERENCES

1. Copyright Act of 1976, 17 U.S.C.A. §§101, 201(b) (West Supp. 1993).
2. 490 U.S. 730 (1989).
3. *Public Affairs Assocs. Inc. v. Rickover*, 177 F. Supp. 601, 604 (D.D.C. 1959), Rev'd on other grounds, 284 F.2d 262 (D.C. Cir. 1960), Vacated for insufficient record, 369 U.S. 111 (1962).
4. *Marshall v. Miles Laboratories* 647 F. Supp. 1326 (N.D. Ind. 1986) [technical article written at home at employee's initiative held to be owned by employer].
5. *Schmid Bros. Inc. v. W. Goebel Porzellanfabrik KG*, 589 F. Supp. 497 (E.D.N.Y. 1984) (drawings used to produce porcelain figurines created by nun held to be owned by her despite convent support of all her activities); *Public Affairs Associates Inc. v. Rickover*, 284 F. Supp. 444 (D.D.C. 1967) [speeches of Vice Admiral written and delivered on his own time but duplicated on government equipment held to be owned by him].
6. 805 F. Supp. 1312 (E.D. Va. 1992).
7. *Id.* at 1318-19.
8. *Id.* at 1319-22.
9. 808 F. Supp. 1238 (D.S.C. 1992).
10. Restatement (Second) of Agency §§228, 229 comment b (1958)
11. *Id.* at 1243.
12. *Id.* at 1244.
13. Copyright Act of 1976, 17 U.S.C.A. §§201, 204 (West Supp. 1993).
14. 980 F.2d 857 (2d Cir. 1992).
15. *Id.* at 862.
16. *Id.* at 862-63.
17. *Id.* at 864.
18. *Id.* at 863.
19. See e.g., *Marco v. Accent Publishing Co. Inc.*, 969 F.2d 1547 (3d Cir. 1992).
20. See e.g., *Respect Inc. v. Committee on the Status of Women*, 815 F. Supp. 1112 (N.D.Ill. 1993).
21. 831 F. Supp. 295 (S.D.N.Y. 1993).
22. See also *Schiller & Schmidt Inc. v. Nordisco Corp.*, 769 F.2d 410 (7th Cir. 1992).

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
- ALL GERBER FORMATS

- FULL TIME MODEM
- GERBER PHOTO PLOTTING

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

Express

Circuits

1150 Foster Street • P.O. Box 58
 Industrial Park Road
 Wilkesboro, NC 28697

Quotes:
 1-800-426-5396
 Phone: (910) 667-2100
 Fax: (910) 667-0487

DEPARTMENTS

44 Firmware Furnace

54 From the Bench

62 Silicon Update

68 Embedded Techniques

77 Patent Talk

84 ConnecTime

FIRMWARE FURNACE

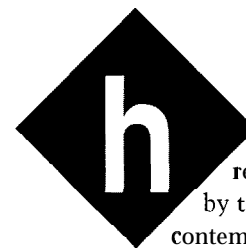
Ed Nisley

Bringing the '386SX Project's Graphic LCD Panel to Life



If you've
been
ready for
some

action with Ed's graphic
LCD panel, this month's
installment gets you
started with routines
that will give you plenty
to watch plus will
exercise your system's
extended memory.



Have you ever repaired something by taking it apart, contemplating the pieces, then putting them back together again? It's hard to believe a simple laying on of hands can fix an inanimate object, but quite often that's what happens.

Textbooks relate such problems to contact corrosion, random glitches, metastability, software errors, and similar maladies. Experienced engineers and technicians know differently. "That widget just needed some attention," they'll say with a smile and go on to the next problem. To hear them talk, you'd almost think it was alive.

In the previous two columns, I covered the Graphic LCD Interface's hardware and some test routines to exercise the circuitry. This month I'll describe routines that manipulate individual dots on the panel and, as an added bonus, put your '386SX system's extended memory to good use.

While the code isn't alive yet, we're getting there.. .

PUTTING LIFE TO WORK

Graphic LCD panels have a voracious appetite for data: testing the drawing routines means exercising a quarter-million dots in that 640x400 array. Just writing and reading a byte

Listing 1—One of the attractions of Conway's Game of Life is the simplicity of the algorithm: these few lines determine the changed cells in the next generation. Because the graphic LCD panels are so large, however, the complete computation can take tens of seconds even on a fast processor. For our purposes this is fast enough, but you can have fun optimizing the code beyond recognition.

```

for (Row=0; Row<NumRows; Row++) {
    ByteToLEDs(Row);           /* show some progress */
    outpw(SYNC_ADDR,0x80);     /* for scope timing */
    for (Col=0; Col<NumCols; Col++){
        Neighbors = LCDGetDot(Row-1,Col-1) +
                    LCDGetDot(Row-1,Col) +
                    LCDGetDot(Row-1,Col+1) +
                    LCDGetDot(Row,Col-1) +
                    LCDGetDot(Row,Col) +
                    LCDGetDot(Row,Col+1) +
                    LCDGetDot(Row+1,Col-1) +
                    LCDGetDot(Row+1,Col) +
                    LCDGetDot(Row+1,Col+1);
        if (LCDGetDot(Row,Col)){
            if ((Neighbors < 2)|| (Neighbors > 3)){
                LCDSetDot(Row,Col,0);

            else {
                if (Neighbors == 3){
                    LCDSetDot(Row,Col,1);
                }
            }
        }
    }
    outpw(SYNC_ADDR,0x00);
}

```

correctly doesn't mean that the dots appear at the right place on the panel, so you must actually look at the results to see if it's working. But who wants to watch a test pattern!

Conway's venerable Game of Life is the great-to-the-nth grandfather of today's Artificial Life creations. It produces an easily recognizable set of patterns, exercises the whole LCD panel, is fun to watch, and is falling-off-a-log easy to code.. if performance isn't an issue, anyway.

The playing field is a rectangular grid of cells, each of which may be either "alive" or "dead." The field should be infinite, but typical implementations surround the visible part of the field with permanently dead cells or join the four borders to simulate a torus. The initial cell contents uniquely determine the course of the game, so Life isn't interactive.

Play proceeds by generations and the state of all the cells changes at one instant at the end of each generation. The state of each cell in the next generation depends on its current state and the state of its eight immediate neighbors.

A dead cell remains dead unless exactly three of its neighbors are alive, in which case it springs to life in the next generation. A live cell remains alive if exactly two or three of its neighbors are alive, otherwise it dies of either loneliness or overcrowding. Variations on those rules are possible, but tend to produce less interesting patterns.

Listing 1 shows the few lines of code needed for this algorithm in its most basic form. A little arithmetic shows how much computation is needed for a field filling a 640x400 LCD panel. There are $9 \times 640 \times 400 = 2.3 \times 10^6$ LCDGetDot calls per generation!

Relatively few cells change state after the first few generations, so the LCDSetDot calls don't influence the overall time. I used two separate fields, one in system RAM and the other in the LCD Refresh buffer, but the few milliseconds devoted to copying the "new" dots into the "old" buffer at the end of each generation have no effect, either.

The calculations take about 100 μ s per cell, so computing one genera-

tion takes $100 \mu\text{s} \times 640 \times 400 = 26$ seconds. Watching this is similar to studying the life cycle of glaciers in real-time. Smaller panels are faster, but remain in the icicle-growth competition. This is obviously a program where optimization pays off handsomely!

Michael Abrash presented a Game of Life Optimization Challenge in the June '92 PC *Techniques*, with the results appearing in the Dec '93 issue. The two winning entries run at 125 ns/cell(!) on a 33-MHz '486 and their source is a wonder to behold. Before you start tweaking my simple code, read those two articles.. and weep.

Because I was more interested in LCD panels than the Game of Life, I decided to cheat. I added a smidge of code to store each generation in extended memory, then play them back as a sort of slide show. You'll get the details after we cover the LCD routines.

DOING DOTS

The Graphic LCD Interface maps the LCD's dots into a 32K chunk of the PC's address space. From the CPU's viewpoint, the LCD panel is just a relatively slow RAM with the useful side effect that the bits are visible to the naked eye. Unfortunately, each panel has its own dot layout, so we need unique code to find each dot in RAM.

Finding a dot requires two steps: selecting the RAM byte holding the bit, then isolating the bit within that byte. Obviously, both steps depend on the dot's row and column address. To simplify the high-level code, rows and columns start at 0 in the upper-left corner of the panel even though the LCD panel documents number them differently. Other mappings are certainly possible, so if an upside-down or backwards mapping suits your needs, go for it!

As you saw last month, the top row of dots on the LCD panel *does not* come from the first group of bytes in RAM. The Graphic LCD Interface produces a Frame Sync pulse when it resets the address counters to zero, but the panel expects Frame Sync to follow the first row. As a result, the dots

starting at address 0000 appear on the *second* panel row.

The quick and easy solution for this appears in Listing 2. The starting address of each row is stored in a table, which, for the LG64AA44D, has 400 entries. The first entry points to the bytes starting at address $199 \times 160 = 76C0$, while the second entry holds address 0000. The dot drawing routines use the table to find out where the rows start, so the scrambled addresses unscramble the panel layout to put Row 0 at the top of the panel where it belongs.

I fill the table as part of the LCD initialization function. Entries 0 and 200 are special-cased at the top of the loop, while the remaining 2 through 199 entries are a simple ascending sequence. I suspect you could define and fill the table using some assembler macro magic, but this way is easy enough.

The LCDMakeAddr function in Listing 3 converts a dot's row and column address into a byte address and a bit shift amount. The byte address depends on both the row, which specifies the lookup table entry, and the column, which specifies an additional number of bytes from the start of the row.

These LCD panels pack several dots in each byte, so I use the bit shift value to move the selected bit to the low-order bit or vice versa. For the LG64AA44D, the shift amount depends only on the column, but the LM2 15, which I'll describe later, is more complex.

Homework assignment: step through the code with a pencil and paper to see how the algorithm works. Draw a picture of the LCD panel's bit layout, indicate the bit addresses at the start of each row, then verify that the code really does produce the right answers. You won't believe it until you do...

Listing 4 pulls all of this together to write a single dot. As you saw in Listing 1, LCDSetDot is called directly from the main C routine, so its register and stack usage must match C's expectations. This used to be fiendishly tricky, but current assemblers include several high-level directives to

Listing 2-A lookup table simplifies finding the RAM address of a particular dot by holding the starting address of the corresponding row. This code excerpt defines and loads the 400 table entries needed for a Matsushita LG64AA44D 640x400-dot panel. The data for the lower half of the panel is located in the high-order nybble of the same bytes displayed on the upper half, so the second half of the table (rows 200-399) contains the same addresses as the first half (rows 0-199).

```

LCDNAME EQU "Matsushita EDM LG64AA44D"
NUMROWS = 400
NUMCOLS = 640
COLCLOCKS = 160
COLMODULUS = 4
TOTALCLOCKS = 32000
ROWMODULUS = NUMROWS/2
FILLVALUE = 000h

RowStarts DW NUMROWS DUP (?) ; row starting addresses

MOV [RowStarts],(ROWMODULUS-1)*COLCLOCKS
MOV [RowStarts+2*ROWMODULUS],(ROWMODULUS-1)*COLCLOCKS

MOV DI,(OFFSET RowStarts) + 2
MOV CX,ROWMODULUS-1
MOV AX,0 ; row 0 = second entry
PUSH DS ; set up STOSW segment
POP ES

@@1: MOV [DI+2*ROWMODULUS],AX ; second half of table
STOSW ; first half
ADD AX,COLCLOCKS ; offset of next row
LOOP @@1

```

ease the task. I won't go into the details here, but it's a lot easier than counting bytes and tweaking registers on your own.

Recall that LCDMakeAddr returns a byte offset in DI rather than a complete segment:offset address.

LCDSetDot adds the far pointer in pNewBuff to DI to create the final address in ES:DI. You can aim LCDSetDot at the actual LCD refresh buffer RAM or a separate working buffer in system RAM just by changing pNewBuff.

Listing 3-This routine for an LG64AA44D panel converts a dot's row (in BX) and column (in AX) address into the offset of a byte from the start of the buffer (in DI) and the shift amount (in CL) needed to align the bit in Bit 0. The high nybble of each byte stores the dots for rows 200-399, the low nybble has rows 0 through 199, and thus the shift amount depends on which half of the panel holds the dot.

```

PROC LCDMakeAddr

ADD BX,BX ; make word table index
MOV DI,[RowStarts+BX]

DIV [BYTE ColModulus]
XOR AH,03h ; flip shift direction
MOV CL,AH ; save remainder for shifting
MOV AH,0
ADD DI,AX ; DI points to the byte

CMP BX,2*ROWMODULUS ; rows >=200 use high nybble
JB @@1
ADD CL,4 ; ... so shift into it

@@1: RET

ENDP LCDMakeAddr

```

Listing 4-This routine sets or clears a dot on the LG64AA44D LCD panel. The global variable pNewBuff contains a farpointer to the target buffer, which may be the LCD panel or a chunk of system RAM. Depending on your application you may want to include the target address as one of the function parameters. This code can be called directly from the main C program because the first four lines handle the inferlanguage register and stack conversions.

```

PUBLIC C LCDSetDot
PROC LCDSetDot
ARG Row:WORD,Col:WORD,Value:WORD
USES ES,DI

MOV AX,[Col]
MOV BX,[Row]
CALL LCDMakeAddr

LES AX,[pNewBuff] ; get buffer base pointer
ADD DI,AX ; ... update byte pointer

MOV BX,0101h ; set up bit mask
SHL BX,CL
NOT BL ; BL = AND mask. BH = OR mask

TEST [Value],0001h ; what do we want?
JNZ @@2 ; nonzero means set the bit
AND [ES:DI],BL ; ... clear it
JMP SHORT @@3

@@2:
OR [ES:DI],BH ; ... set it

@@3:
RET

ENDP LCDSetDot

```

Normally you would include the buffer base address in the arguments you feed to LCDSetDot. I used a global variable to reduce the amount of stack shuffling, but I agree that globals can cause problems in a real application. As always, use this code as the basis for your own rather than The Final Word on style.

BX holds the bit masks needed to set or clear the selected bit. Although only one instruction touches the target byte, the CPU actually makes two memory accesses: one to read the current dots and another to write the updated byte. This is as fast as you can do it, but the LCD RAM on the ISA bus is achingly slow compared to DRAM on the system board.

Listing 5 shows how LCDGetDot tests an LCD bit. In this routine the global far pointer pOldBuff sets the buffer's RAM address. The shift amount in CL now moves the selected bit into the LSB of AX, where it's masked off to make a boolean value for the calling routine.

LCDGetDot and LCDSetDot use two different global pointers because the Game of Life's field must remain unchanged until all the old cells are tested. I aimed pNewBuff at the LCD refresh buffer so you could watch the dots appear, but you could put them in system RAM. Just copy them all to the LCD RAM at the end of each generation to reveal the new field.

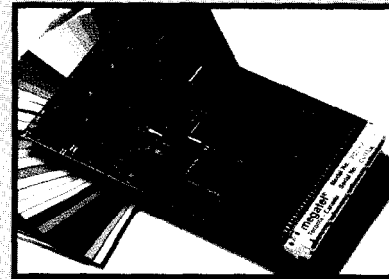
The main loop calls LCDSetDot only when a cell changes, which means that new field must contain the old cells at the start of each generation. I used the C memcpy library function to copy the entire 32K-byte buffer at the end of each generation. Compared to the time needed to compute a generation, memcpy is essentially free.

VARIATIONS ON A THEME

Even through the LG64AA44D panel has a relatively complex dot layout, the code is still fairly simple. Rather than list the routines for the 640x200 DMF65 1 and the TLY-365

Embedded PC

"The new megatel
PC/+v offers on-board
VGA in a small rugged
form factor."



PC/+v

On-board Features Include:

- 16 MHz V-40 CPU
- 640KByte User DRAM
- On-Board VGA Video/LCD display controller
- Power Consumption under 2 watts, +5 volt only operation
- 3 RS-232 Serial Ports
- Floppy, SCSI, Keyboard Ports
- Parallel, Printer Port
- PC Compatible BIOS
- Real Time Clock
- Only 4" x 6" (100 x 160mm)
- ISA and PC/104 Bus Options

For a list of our International Distributors,
please contact our head office at:

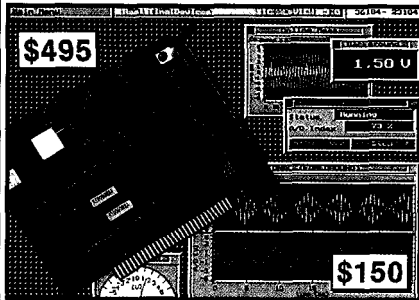
megatel computer corp.
125 Wendell Ave.
Weston, Ont. M9N 3K9 Canada
Fax: (416) 245-6505

(416) 245-2953

megatel®

Data Acquisition and Control Without Compromise

Programmable Scan & Burst
RTD's Advanced Industrial Control
boards set a new performance
standard for general purpose
industrial & laboratory applications.



ADA2210 with SIGNAL*VIEW
A cost-effective solution

ADA2210 features:

- 125 kHz XT throughput
- 16 SE. / 8 DIFF 12-bit analog inputs
- $\pm 5, \pm 10, 0 \rightarrow +10V$ selectable input range
- programmable auto channel scan
- programmable burst mode
- software & external triggers, pacer clock
- on-demand DMA transfer
- programmable gain: 1/2/4/8 (1/10/100/1000)
- 13 cascadable 16-bit counters
- 16 programmable digital I/O lines
- two 12-bit analog outputs, selectable range
- rtdLinX Universal TSR DOS driver
- Labtech Notebook driver

HARDWARE SOLUTIONS

Our AIC family also includes AT analog I/O boards, +5V only A/D & control boards for portable PCs, 4-20 mA current loop outputs, and opto-22 compatibility.

SOFTWARE SOLUTIONS

Select the power and performance of SIGNAL*VIEW™ or SIGNAL*MATH™ from our library of application programs for monitoring, data acquisition and analysis, control, DSP and 3D graphics.

For more information on these and other
ISA bus and PC/104 products,
call, write or fax us today!

Place your order now and receive
SIGNAL*MATH and
SIGNAL*VIEW for
\$195!



Real Time Devices, Inc.

P.O. Box 906
State College, PA 16804
(814) 234-8087 ■ Fax: (814) 234-5218

121 panels, I'll aim you at the BBS for the source code if you need it. I decided not to do the 640x400 Sharp panel because the inverters are so hard to find, but if you have one I'll be glad to kibitz.

The 480x128 LM215 merits some discussion because it is so peculiar. Again, check the BBS for the actual code. Each of the four bits in the low nybble drives a separate quadrant: the panel has four 240x64 subpanels. It uses a 960-ns Dot Clock, so the dots must be duplicated in successive even- and odd-addressed bytes in the LCD refresh buffer to present stable data to the panel.

The LM215's row address table has 128 entries, corresponding to the 128 visible rows. The code I used to fill it resembles that shown in Listing 2. I put only the even addresses in the table and modified the LCDGetDot and LCDSetDot functions to generate the corresponding odd addresses as needed.

In the LCDMakeAddr function, the bit shift amount depends on the dot's quadrant, so I used the '386 SETA E instruction to generate a boolean value from the CMP instructions rather than interject several branches. The byte offset added to the table entry is the same for the first and last 240 columns in each row. SETA E doesn't exist on 8088 or 80286 CPUs, which means you must tweak the code if you're not using a '386.

The LM215 LCDSetDot function updates both bytes with the new dot value. Apart from that, it's basically the same as the code for the other panels because LCDMakeAddr handles the peculiar bit and byte addressing.

The LCD initialization routines for all these panels disable blinking by gating a constant zero to the LCD Data Mux (U54). While there are good reasons for blinking dots, the Game of Life doesn't need any, so it's left as an exercise for the interested reader.

STASHING SLIDES

I used the Game of Life to produce dots because it is far more interesting than a bland test pattern, but waiting half a minute for each update is a strain on even the most placid attention span. Rather than get involved

with code optimization, I decided to store successive generations in RAM and then play them back quickly. While the slide show repeats after a while, that's OK in this application because you can study problems as they crop up over and over again.

So far in this series our '386SX CPUs have been running in Real mode, so only 20 address bits are active and storage is limited to 1 megabyte. The PC's memory addressing restrictions [discussed in Issues 36 and 37] limit contiguous RAM to 640K starting at address 00000. After loading a program, allocating a few work buffers, and leaving room for a decent far heap area, there's only a few hundred kilobytes left over.

In contrast, when a '386 CPU runs in Protected mode, it uses 32 address bits to reach 4 GB of memory. The '386SX has only 24 address pins and is thus limited to only 16 MB, but that's significantly more than 640 KB. In PC systems, the first megabyte of the Protected mode address space is devoted to the familiar Real mode RAM and I/O devices, with the remaining storage appearing as a contiguous expanse starting at the 1-MB line.

This RAM is called "extended" memory to distinguish it from "expanded" memory that is page-mapped into the lower 1 MB. The (now largely irrelevant) LIM 4.0 spec defines the operation of a standard DOS Expanded Memory Manager that controlled actual hardware. The EMM386 DOS device driver controls extended memory and can simulate expanded memory with no additional hardware.

Your '386SX system board probably has a multiple of 2 MB of RAM because the CPU addresses memory in two-byte chunks. Typical (cheap) SIMMs are one byte wide [plus a parity bit] so they must be installed in pairs. You need at least 1 MB to run the system, and two 1 MBx9 SIMMs are cheaper than four 256Kx9 parts. QED. In any event, you have more memory available beyond 1 MB than you do below it!

Although the CPU must run in Protected mode to access expanded memory, we don't need a complete

Listing 5—This routine tests a dot on the LG64AA44D LCD panel and returns its state as a C Boolean value. If the row or column lies outside the LCD's boundaries the return value is always zero. While this simplifies the higher-level code for the Game of Life, it may not be the right action for all programs!

```

PUBLIC C LCDGetDot
PROC LCDGetDot
ARG Row:WORD,Col:WORD
USES ES,DI

XOR AX,AX ; set up zero return value

MOV BX,[Col] ; verify column
TEST BH,80h ; negative?
JNZ @Done ; . yes
CMP BX,NUMCOLS ; too big?
JAE @Done ; yes

MOV BX,[Row] ; verify row
TEST BH,80h ; negative?
JNZ @Done ; . . . yes
CMP BX,NUMROWS ; too big?
JAE @Done ; . . . yes

MOV AX,[Col] ; set up column number
CALL LCDMakeAddr

LES AX,[pOldBuff] ; get buffer base pointer
ADD DI,AX ; . . . update byte pointer

MOV AL,[ES:DI] ; . . . fetch the byte
SHR AL,CL ; skid our bit into LSB
AND AX,0001h ; isolate it

@@Done:
RET

ENDP LCDGetDot

```

Protected mode program. The system BIOS includes functions to copy blocks of data to and from the 1 MB available to Real mode programs, so we can create the data as usual and dispose of it through a BIOS call.

The first step is finding out how much extended memory is available on the system. Listing 6 shows the code needed to access BIOS Int 15 (hex, remember) Function 88 (likewise), which returns the extended memory size in kilobytes in register AX. Because AX is only 16 bits wide, the maximum memory this function can handle is merely 64 MB. . . enough for the '386SX, but painfully cramped in these days of system boards with 256 MB (or more!) of RAM.

The various references disagree as to whether Function 88 clears the carry flag to indicate success. In my system, carry is set even though AX contains the correct value, so I wrote the code to weed out some obviously

bogus return values. If it misbehaves on your system, write a few test cases and report back on the BBS.

I turn bit 0 of the printer port at SYNC_ADD R on before invoking the software interrupt and off when it returns, so if it stays on, you know your system has problems. This will certainly happen on 8088 systems and will probably happen on oddball '286 systems because the BIOS functions may not be quite compatible.

Function 88 reads the extended memory size from the CMOS clock/calendar RAM, where it was stored during the BIOS power-on tests after each reset. Some systems store this value as part of a manual setup and complain if it doesn't match the actual amount of RAM found during POST. In any event, the return value should be 1 MB less than the total amount of RAM installed in your system.

BIOS Int 15 Function 87 copies a block of data between any two loca-

tions in memory. The process is straightforward: enter the source and destination addresses in a table in RAM, specify the number of 16-bit words to move in CX, aim ES:SI at the table, and issue the software interrupt. Unfortunately, the references disagree on exactly what's needed in the table in addition to the addresses, if anything.

It turns out the table is actually the Global Descriptor Table used when the CPU enters Protected mode. Listing 7 shows the structure of the table entries: even though the code runs on a '386SX CPU, the BIOS expects an 80286 GDT for compatibility with the original PC/AT. The two addresses form a pair of segment descriptors that the CPU uses to access the data blocks.

Listing 8 is the code that copies the Life fields into extended memory. The GDT memory addresses are 24-bit linear values instead of the segment: offset pairs we all love to hate. I fill in the segment length and access flag GDT entries even though some of the references imply the BIOS will do this.

I decided to copy the entire 32K-byte block of storage, which makes the extended memory addresses easy to decode. If you apply data compression before storing the frames, remember to tweak CX and step the addresses by the appropriate amount.

The two out p w functions mark the CPU's journey into Protected mode by blinking bit 1 on the printer port. As before, if that bit stays on you know your system is lost in hyperspace. My 33-MHz '386SX system takes about 32 ms to transfer a 32K-byte block, which works out to 1 MB per second. The transfer is entirely within the system board RAM, so the usual ISA bus slowdown doesn't come into play, but the switch to and from Protected mode adds considerable overhead.

The BIOS disables hardware interrupts during the transfer because it doesn't have access to any Protected mode interrupt handlers. As a consequence, this function isn't useful if you're trying to keep up with fast serial data or other high-speed interrupts. There are several possible error

return codes that I simply ignore here, but you must deal with them in a real application.

After extended memory is full of Life generations, the code invokes Function 87 again to copy the data back into memory below 1 MB. One of the FDB's DIP switches controls whether the data is then copied into the LCD buffer using `memcpy` or a nested pair of C loops so you can see which affects the display least. A hardcoded 500-ms pause between each slide gives you time to think about what's going on.

One megabyte of extended memory holds 32 generations of 32 KB each, which works out to 16 seconds of Life. My system has 6 MB (5 MB above the 1 MB line) and can store 160 generations, which makes for a longer show. The low-order byte of DIP switches sets the initial random number seed, so you have a choice of 25.5 different slide shows.

Because 200-line panels use only the low-order nybble of each byte, you could store twice as many generations by applying a no-brainer compression scheme. The Life field has many dead cells after the first few generations, so you could probably pick up another 50% by run-length encoding what's left... have fun!

Don't be surprised to find yourself rooting for clumps of dots as they expand, contract, and consume each other. Life is like that..

RELEASE NOTES

The BBS files this month include the source code and binary files for the four panels mentioned in the text. The code is written in Borland C and processed through Paradigm's Locate utility, so copy the BIN files onto a diskette with the boot loader from Issue 41 and fire up your LCD.

The code unique to each panel is in an assembler file with the panel's name. The MAKEFILE produces a similarly named binary on drive A, so have a diskette with the boot loader ready when you recompile the code. If you want to add another panel, use one of the existing files as a template, update the MAKEFILE, and you're all set.

Listing 6-BIOS *Int 15* Function 88 returns the extended memory size in KB. This memory starts just beyond the 1 MB of memory addressable in Real mode. Because AX has only 16 bits, the function can report only the first 64 MB of extended memory!

```
regs.h.ah = 0x88;
outp(SYNC_ADDR,0x01); /* mark the query */
int86(0x15,&regs,&regs);
outp(SYNC_ADDR,0x00);

XMemSize = regs.x.ax;
printf(" BIOS reports %u KB\n",XMemSize);

if ((XMemSize >> 8) >= 0x0080){
    puts("... on a 386SX you can't have that much, so it's ignored");
    XMemSize = 0;
}

XMemLimit = (long)(XMemSize / FRAMESIZE);
/* truncate limit to a */
XMemLimit *= (long)FRAMESIZE * 1024L; /* multiple of the slide */
XMemLimit += ONE_MEG; /* above the line */
printf("Room for %u frames of %u KB each between %081X and %081X\n",
    XMemSize/FRAMESIZE,FRAMESIZE,ONE_MEG,XMemLimit-1);
```

You also get my version of Paradigm's C0NSOLE.C file. I replaced their demo code with BIOS serial code to support simple console I/O. If you're using Locate, you must also set up the FARH EAP constant (I used 0x8000) to allocate a decent-sized far heap for the program's work buffers.

Dave Dunfield politely pointed out that the Micro-C problem I'd mentioned in Issue 40 was entirely of my own making. The Micro-C interrupt handler should load DS with its own data segment, then save `?temp`. If the interrupted routine shared the same data segment, the old `?temp` is saved and restored correctly. If the DS

values are different the interrupt handler is saving and restoring its own `?temp`, but with no ill effect.

He mentioned that the compiler uses `?temp` in certain complex operations, so you can't just throw it out. My examples were simple enough that they didn't use it, but real code would have trouble. ☹

Ed Nisley, as Nisley Micro Engineering, makes small computers do amazing things. He's also a member of the Computer Applications Journal's engineering staff. You may reach him at ed.nisley@circellar.com or 74065.1363@compuserve.com.

Listing 7-BIOS *Int 15* Function 87 copies memory between any two addresses in RAM, specified as 24-bit linear values rather than the familiar segment/offset pairs used below 1 MB. You must fill out parts of the Global Descriptor Table Function 87 will use when entering Protected mode, although the references disagree on exactly what the GDT must contain. The GDT is an array of the structures shown here. Even though the CPU is a '386SX, the BIOS uses a '286 format GDT for compatibility with the original PC/AT.

```
#pragma option -a /* structure must be packed */
typedef struct{
    WORD SegLimit; /* size of segment in bytes */
    WORD SegBaseLow; /* low two bytes of linear address */
    BYTE SegBaseHigh; /* high byte of linear address */
    BYTE SegFlags; /* 9B = code, 93 = data, 91 = RO data */
    WORD Reserved; /* reserved, must be zero */
} GDT_286
#pragma option -a. /* restore default setting */

#define MK_LINEAR(fp) (((long)FP_SEG(fp) << 4) + (long)FP_OFF(fp))
GDT_286 MoveGDT[6]; /* GDT for BIOS Move Block operation */
```

Listing 8—In addition to the (obvious/y) necessary source and destination addresses in the GDT, this code fills in the segment length and access flags even though they're not required on my system. Bit 1 on the printer port marks the CPU's journey through Protected mode; if the LED remains lit, your CPU fell off the tracks. The code to copy data from extended memory is similar: just interchange the addresses.

```
memset(MoveGDT,0,sizeof(MoveGDT));          /* set up zeros */

MoveGDT[2] SegLimit   = FRAMESIZE*1024;
MoveGDT[2] SegFlags   = 0x93;
MoveGDT[2] SegBaseLow = (WORD)MK_LINEAR(p01dBuff);
MoveGDT[2] SegBaseHigh = (BYTE)(MK_LINEAR(p01dBuff) >> 16);
MoveGDT[3] SegLimit   = FRAMESIZE*1024;
MoveGDT[3] SegFlags   = 0x93;
MoveGDT[3] SegBaseLow = (WORD)XMemAddr; /* target */
MoveGDT[3] SegBaseHigh = (BYTE)(XMemAddr >> 16);

segread(&sregs); /* get current seg regs */
regs.h.ah = 0x87; /* move block */
regs.x.cx = FRAMESIZE*(1024/2); /* number of words */
regs.x.si = FP_OFF(MoveGDT); /* offset of GDT */
sregs.es = FP_SEG(MoveGDT); /* segment */

outp(SYNC_ADDR,0x02); /* mark the move */
int86x(0x15,&regs,&sregs); /* shazam! */
outp(SYNC_ADDR,0x00);

printf(" stored at %081X",XMemAddr);

XMemAddr += (long)FRAMESIZE * 1024L; /* step to next block */
```

SOURCES

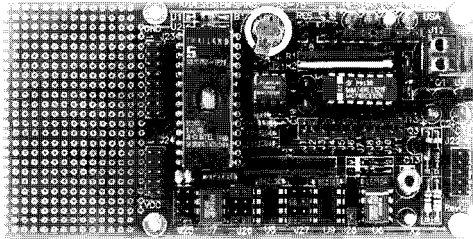
Crystaloid Electronics Company produces a booklet titled "Liquid Crystal Displays: A Primer" that explains how various types of LCD panels work and explores system design tradeoffs. It's intended for OEM customers who must select or specify a panel, so it's not too technical. Contact them at 5282 Hudson Dr., Hudson, OH 44236, (216) 655-2429, fax (216) 6552176.

Pure Unobtainium has the complete Firmware Development Board schematic, as well as selected parts. Write for a catalog: 13 109 Old Creedmoor Rd., Raleigh, NC 27613. Phone or fax (919) 676-4525.

IRS

410 Very Useful
411 Moderately Useful
412 Not Useful

NEW! 87C752 CONTROLLERS



Iota continues its strategy of developing small powerful controllers with its new line using the 87C752, 87C751 and 87C750 microcontrollers. We have also developed our own BASIC-752™ which runs in the 87C752. Several software tools are available to ease the development process.

We offer:

- I²C-bus peripherals
- Complete Development Packages
- Powerful controller BASIC for the 87C752
- A BASIC-752 Simulator

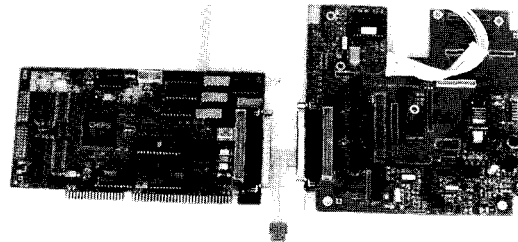
Voice: **702.831.6302** • Fax: **702.831.4629**

Iota Systems, Inc.

POB 8987 • Incline Village, NV 89452-8987

Flat panel interfacing made simple!

Just install the modules and you are ready for action!



SVGA Flat Panel Controller, Interface Card and Touch Screen Module

Super VGA Controller

OEM price starts at \$289

- + IBM-AT 16 bit bus compatible
- + Supports LCD, EL, Plasma, **STN** and **TFT** Panels
- + Supports panel resolution up to 1,280 x 1,024
- + 256K VRAM & 256K x 4 Frame Accelerator or 512K Opt.
- + Optional touch screen communication interface

Flat Panel Interface Card

OEM price starts at \$79

- + Supports 640 x 480 and higher resolution panels
- + Built in CCFL backlight **inverter** and LCD bias supply
- + Brightness and contrast controls
- + Extended cable length support

Touch Screen and Controller

OEM price starts at \$249

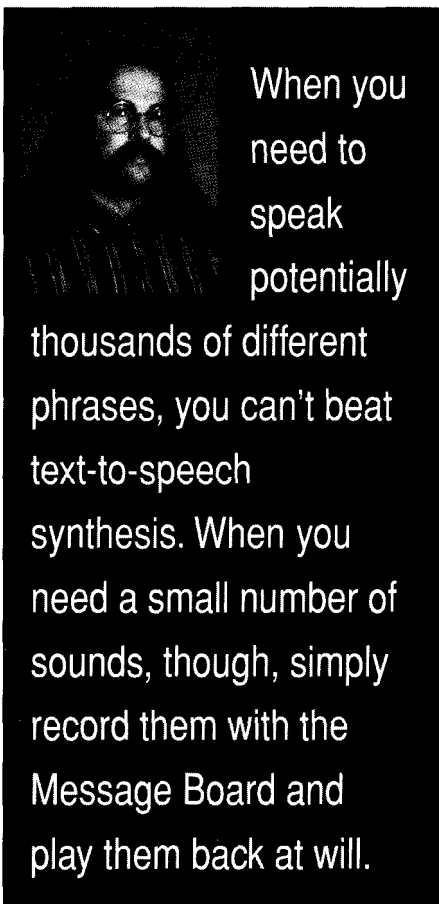
- + Analog Resistive, resolution 1,024 x 1,024
- + 9600 baud **COM1** or **COM2**

Gounder Technologies, Inc. 708-559-9291

115 Avon Rd, Northbrook, IL 60062-1335 fax 708-205-9411

Build the Message Board

An Audio Record/ Playback Unit



When you
need to
speak
potentially

thousands of different
phrases, you can't beat
text-to-speech
synthesis. When you
need a small number of
sounds, though, simply
record them with the
Message Board and
play them back at will.

FROM THE BENCH

Jeff Bachiochi



It's the beginning of the new year as I write this. The snow is knee deep and the oil trucks are as busy as spring's honey bees. I just removed our Christmas tree's skeleton from the house. Not a needle left on it. I remember the devotion to that tree back in December when we picked it out.

Choosing a Christmas tree is always an experience. The festive spirit usually peaks as we drive toward our destination, the local tree farm. After an hour or two of "nope, too short," "that's lopsided," or "this one's missing branches," we have totally lost the spirit of the season and aren't on speaking terms. I know this happens to you; I've heard you arguing over in the next row of trees!

Things turned out a bit different this year. While picking up a few supplies at the local building supply warehouse, the kids noticed they were selling trees. "Let's look," they begged.

"A Canadian orphan," I thought. And there it was, *the* tree. We couldn't believe it: everyone was in agreement. I quickly paid for it while I thought over and over, "Still on speaking terms and the tree was chosen."

The ornaments were hung by the children with care, in hopes that all on their lists would soon be there. Each one of us has our own ornament. I have Star Trek's shuttle craft which repeats a canned message. "Shuttle craft to Enterprise. Spock here. Happy holidays. Live long and prosper."

"I've swept quite a few needles from the floor so far," Beverly prodded.

"Must be the initial shock," I surmised. For the next week it rained

evergreen spikes.

"That tree shouldn't be shedding like this," she needed. "We need a new tree."

"Look, it's only a few days 'til Christmas; it will be fine." I argued, "Besides, look at how well you can see all the ornaments!"

"Everything must be perfect." Her voice became strained, "You know we are having your family over for Christmas dinner this year."

I thought to myself, "Oh yeah, you know how a defective Christmas tree can ruin a party." I could feel the spirit making a run for the door. But it wasn't too late.

After examining the true meaning of Christmas we exchanged no harsh words, only gifts, and had a great Christmas in spite of the naked tree. Its boughs now bare, it sits upright, outside in the snow. Suet and seed decorate its branches. And you know, the birds haven't complained once.

COMPLAINT DEPARTMENT

Many readers of this column were disappointed when the circuitry for the HCS-Voice text-to-speech board was not revealed ("Updating TIM for the HCS II," October '93). We felt that since we couldn't give away the licensed software, having the schematics would only cause distraction. The board is available as a kit. Even though it wasn't covered in detail in an article, its text-to-speech capabilities make it my choice for use with the HCS.

Looking for something a bit simpler you say? This month's project, the "Message Board," uses the new ISD2500 series of analog record/playback devices from Information Storage Devices in a cascade mode to create an expandable messaging system. This is the same technology used in the Shuttle Craft ornament.

ISD2500

The ISD25xx series devices use a 480k-cell nonvolatile analog storage array to provide 45, 60, 75, or 90 seconds of storage depending on the fixed input sample rates of 10.6k, 8k, 6.4k, and 5.33k respectively. DAST (Direct Analog Storage Technology) allows analog data to be written

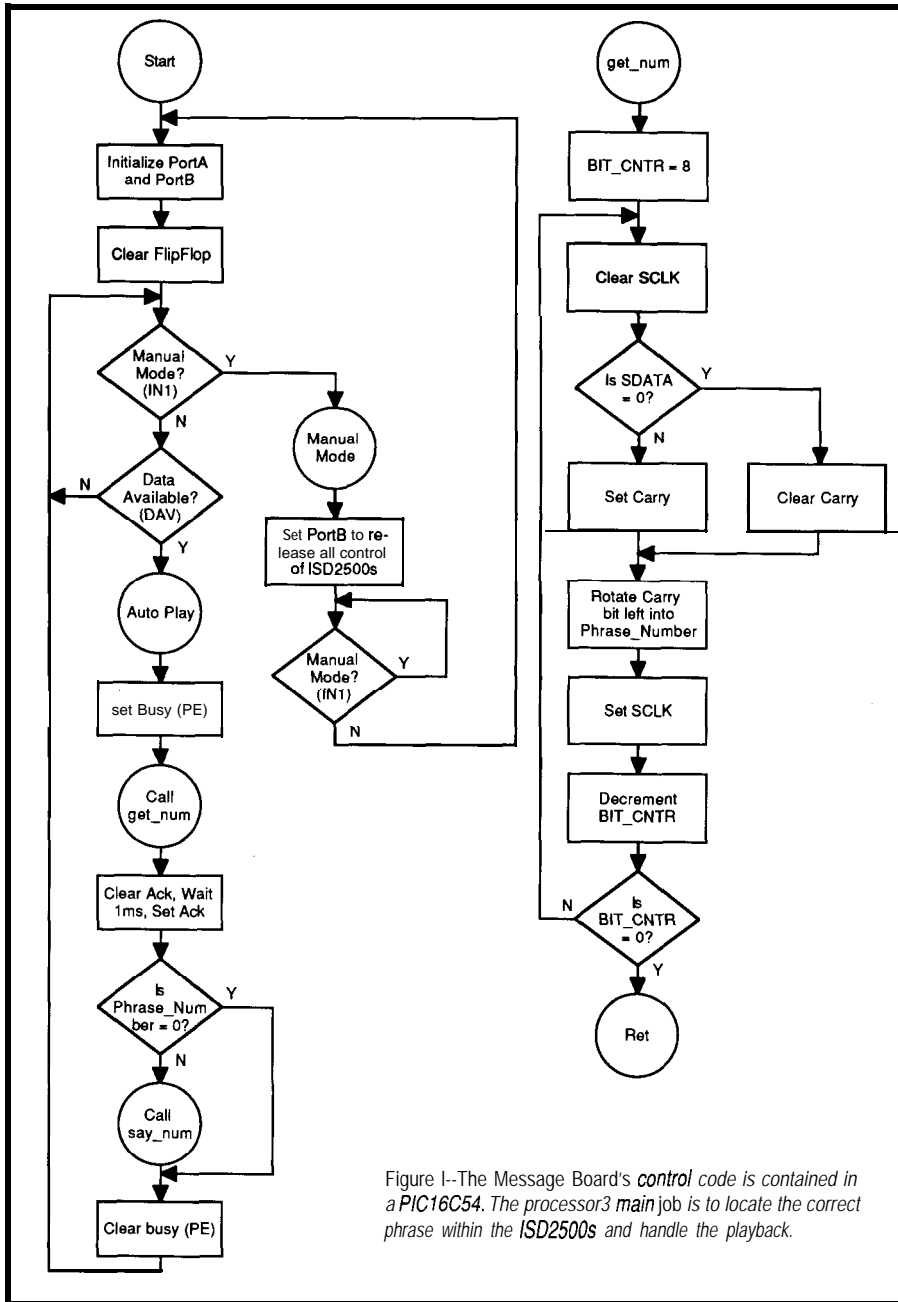


Figure 1--The Message Board's control code is contained in a PIC16C54. The processor's main job is to locate the correct phrase within the ISD2500s and handle the playback.

OPERATIONAL MODES

Message cueing mode (MO) provides the ability to step through recorded phrases without any audio output. I'll be using this mode to allow a micro to index through a message base. The cueing is done in a fast-forward mode which zips through each message at 800 times normal speed.

The Delete EOM marker mode (MI) will concatenate all message phrases into a single message. This is accomplished by eliminating all EOM markers from all previously recorded individual messages and adding one EOM marker to the final message.

Looping mode (M3, M2 is reserved) continues to replay the first message without a pause.

The consecutive addressing mode (M4) prevents the address pointer from being reset upon an EOM marker. This provides the facility to record or playback multiple message phrases within a single chip. This mode, in conjunction with the message cueing mode, is the heart of this solid-state analog record/playback system.

Level-activated mode (M5) allows the playback to be terminated by the rise of the \bullet CE line. (Normal operation would start the message on the falling edge of \bullet CE. At that point, raising \bullet CE has no effect. The message would end following an EOM marker.)

The push-button mode (M6) reduces the external component count to only those necessary to provide a simple push-button solid-state tape recorder. Start/Pause and Stop/Reset push buttons are all that is necessary

directly into an EEPROM cell without A/D conversion and to be read directly from an EEPROM cell without D/A conversion. This means an increase in storage density per word. Data retention can be as much as 100 years while record cycles can exceed 100k times.

On-chip peripherals include a microphone preamplifier with AGC (automatic gain control). During recording, a 5-pole antialiasing filter removes any audio components which approach one-half the sampling frequency. The analog storage array is based on EEPROM floating gate technology and is equivalent to 8 bits

of accuracy. During playback, a 5-pole smoothing filter helps to remove the sampling frequency component and restore the original waveform. Finally, an on-board amplifier will directly drive a 16-ohm speaker with up to 50 mW.

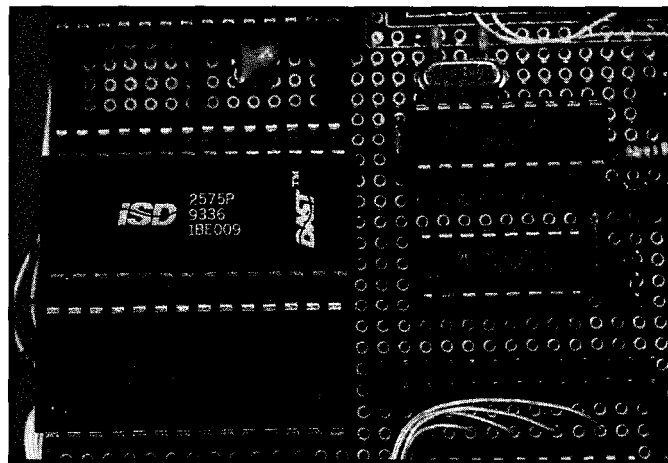


Photo 1—Up to four ISD2500 family chips may be plugged into the Message Board for up to 6 minutes of audio storage.

to provide either record or playback functions. An additional switch will allow record and playback selection. This simple mode is limited to a single message or phrase.

In addition to the previously mentioned modes, direct access addressing can point to the beginning of any of the 600 possible message segments within a single ISD2500 device. If you knew the address of each message, you could point directly to it. However, this is generally not known except in a sound development environment.

See Tom Cantrell's "Talking Chips" article in issue #36 for more information on the older ISD1000 series devices.

A MICRO CONTROLS THE MESSAGE

I wanted this message playback system to be easy to use, so I use a single byte [message number] to indicate one of a possible 255 prerecorded message phrases to play back. To accomplish this, I use a small micro (a PIC16C54) to accept a byte from any parallel port and perform the necessary twiddling on

the ISD2500 devices to fast-forward to the correct message and play it. Notice I used the plural "devices"; the option of using multiple devices increases the overall message storage capability up to a maximum of 6 minutes (four 90-second devices). More on this later.

I/O pins on a small micro are sparse; the 18-pin 16C54 has just 12 I/O lines that can be totally consumed by a parallel connection [including handshaking]. To resolve this, we must bend over backwards to interface with the micro. A parallel-loaded serial shift register requires only two I/O bits to interface as opposed to the eight required for directly reading the byte.

To ensure the byte is not missed (this version of the PIC has no external interrupts), the data's strobe is latched to become a DAV [data available] signal which is periodically scanned by the micro. The micro initiates a busy signal while serially sucking in the parallel port's data, then acknowledges its receipt once the data is digested. See the flowchart in Figure 1 for the micro's code outline.

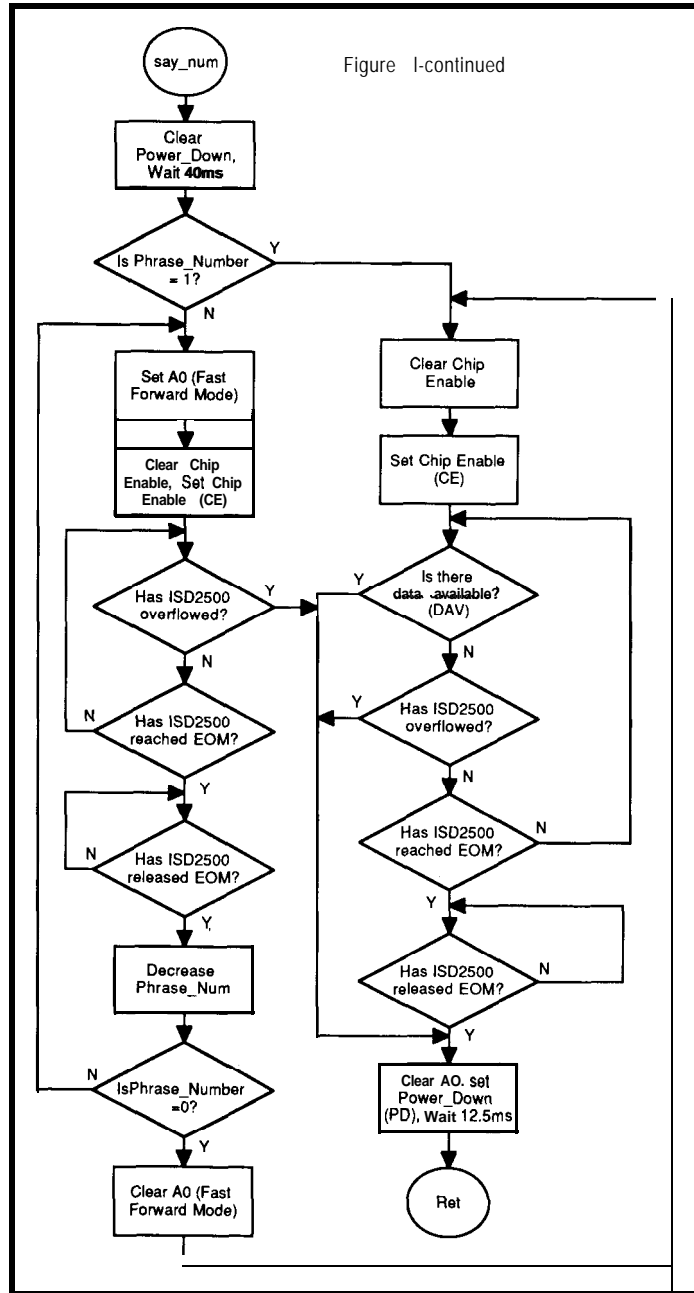
At the other end of the circuit, the micro manages four ISD2500 control functions. The P/R control bit selects the playback/record mode. The PD control bit resets the message pointer and regulates the ISD2500's power-

down mode. The ● CE control bit begins the record/playback function. Finally, the A0 addressing bit selects message cueing (fast-forwarding in the playback mode in combination with *CE). Two additional input bits to the micro are used from the micro. The ● EOM signal indicates the end of a particular message phrase. This is the feedback for cueing up a message phrase. The second input, *OVF, signals the end of the storage device has been reached.

To expand the total storage capabilities of this board, I cascade the ISD2500 devices in a daisy chain fashion. The earlier ISD1000 series devices could not handle multiple messages straddling the boundaries of two devices. Some interesting signal steering goes on when the devices are cascaded. For control, the *CE input of device #1 begins a message. If the message is longer than the storage capability of device #1 then the ● OVF output drops. This output signal from device #1 is used as the *CE input to device #2, enabling the second

device only after the storage capacity of the first has been exhausted. *OVF from the last device is used as feedback to the micro. Individual *EOM outputs from each of the storage devices are OR'ed as feedback to the micro. These two signals keep the micro abreast of the ISD2500's status.

On the audio side, an auxiliary audio input is directed to the speaker output whenever the device is not selected or is in overflow. This is important. Whenever the second device takes over its audio output, which is connected to the first device's auxiliary input, it is passed through the first device's amplifier. Audio from



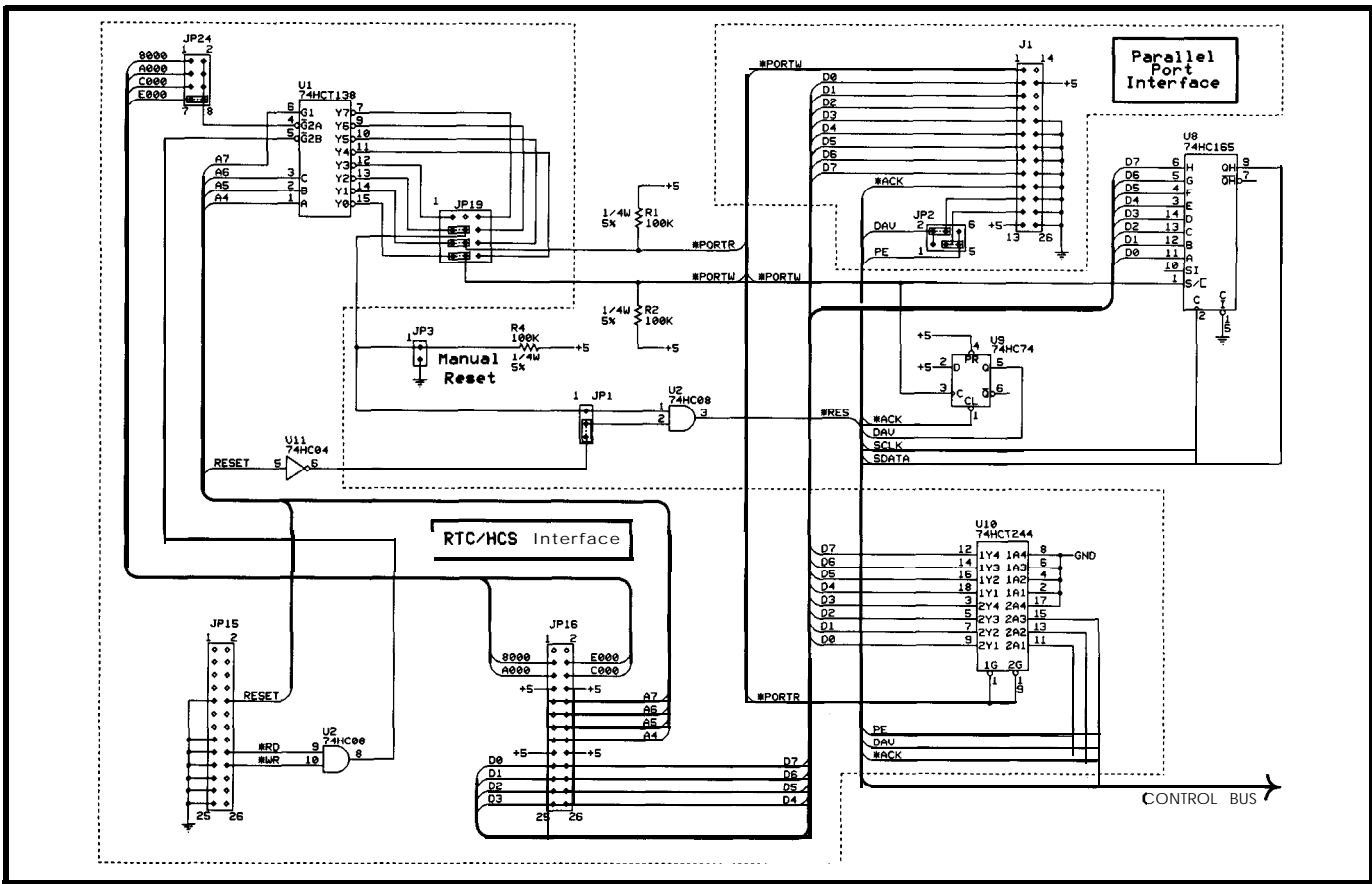


figure 2a—The Message Board can be interfaced to the HCS in the same way as the HCS-Voice board, or may be connected to a parallel port.

Real-Time **Multitasking** with DOS for Microsoft C, Borland C, Borland/Turbo Pascal

Develop Real-Time Multitasking Applications under MS-DOS with **RTKernel!**

RTKernel is a professional, high-performance real-time multitasking kernel. It runs under MS-DOS or in ROM and supports Microsoft C, Borland C, Borland/Turbo Pascal, and Stony Brook Pascal*. RTKernel is a library you can link to your application. It lets you run several C functions or Pascal procedures as parallel tasks. RTKernel offers the following advanced features:

- pre-emptive, event interrupt-driven scheduling
- number of tasks only limited by available RAM
- task-switch time of approx. 6 μ s (33-MHz 486)
- performance is independent of the number of tasks
- use up to 64 priorities to control your tasks
- priorities changeable at run-time
- time-slicing can be activated
- programmable timer interrupt rate (0.1 to 55 ms)
- high-resolution timer for time measurement (1 μ s)
- activate or suspend tasks out of interrupt handlers
- programmable interrupt priorities
- semaphores, mailboxes, and message-passing
- keyboard, hard disk, and floppy disk idle times usable by other tasks
- interrupt handlers for keyboard, COM ports, and network interrupts included with source code
- supports up to 36 COM ports (DigiBoard, Hostless boards)
- supports protocols XOn/XOff, DTR/DSR, RTS/CTS
- full support of NS16550 UART chip
- supports math coprocessor and emulator
- fast, inter-network communication using Novell's IPX or without operating system
- runs under MS-DOS 3.0 to 6.x, DR-DOS, LANs,
- DOS calls from several tasks without re-entrance problems
- supports resident multi-tasking applications (TSRs)
- runs Windows or DOS Extenders as a task
- supports CodeView and Turbo Debugger
- Kernel Tracer for easy debugging
- ROMable
- full source code available
- no run-time royalties
- free technical support by phone or fax

RTKernel-C 4.0 9495 RTKernel-Pascal 4.0 \$ 4 4 5
C Source Code: add \$445 Pascal Source Code: add \$375
International orders: add \$30 shipping and handling.
Mastercard, Visa, check, bank transfer accepted.



On Time
MARKETING

Professional Programming Tools

In North America, please contact:
LEL Computer Systems
20 Canterbury Court • Setauket, NY 11733 • USA
Phone (516) 473-8119 • Fax (516) 331.0706
CompuServe 73313.3177

Outside North America, please contact:
On Time Marketing
Karolinenstrasse 32 • 20357 Homburg • GERMANY
Phone +49 - 40 - 43 74 72 • Fax +49 - 40 - 43 51 96
CompuServe 100140,633



all devices is daisy chained through the first device's speaker output. Likewise, when in record mode, the microphone preamp on device #1 is directed to each of the cascaded devices, resulting in a simple user interface for multiple devices. See Figure 2. Although the ISD2500s will directly drive a speaker, I've added the identical audio output circuitry used on the HCS-Voice board to maintain compatibility with the DTMF board.

SEND A BYTE, SPEAK A PHRASE

A single-byte transfer starts the message playing. Although designed with the same interface as the HCS-Voice, a PC's parallel port can alternatively be used to cue up a message and begin playback. Pre-recorded messages 1-255 are invoked by writing that particular binary value to the Message Board's parallel port. Any message can be interrupted prematurely by sending another message number (the new message begins immediately). Selecting phrase zero stops the speaking altogether.

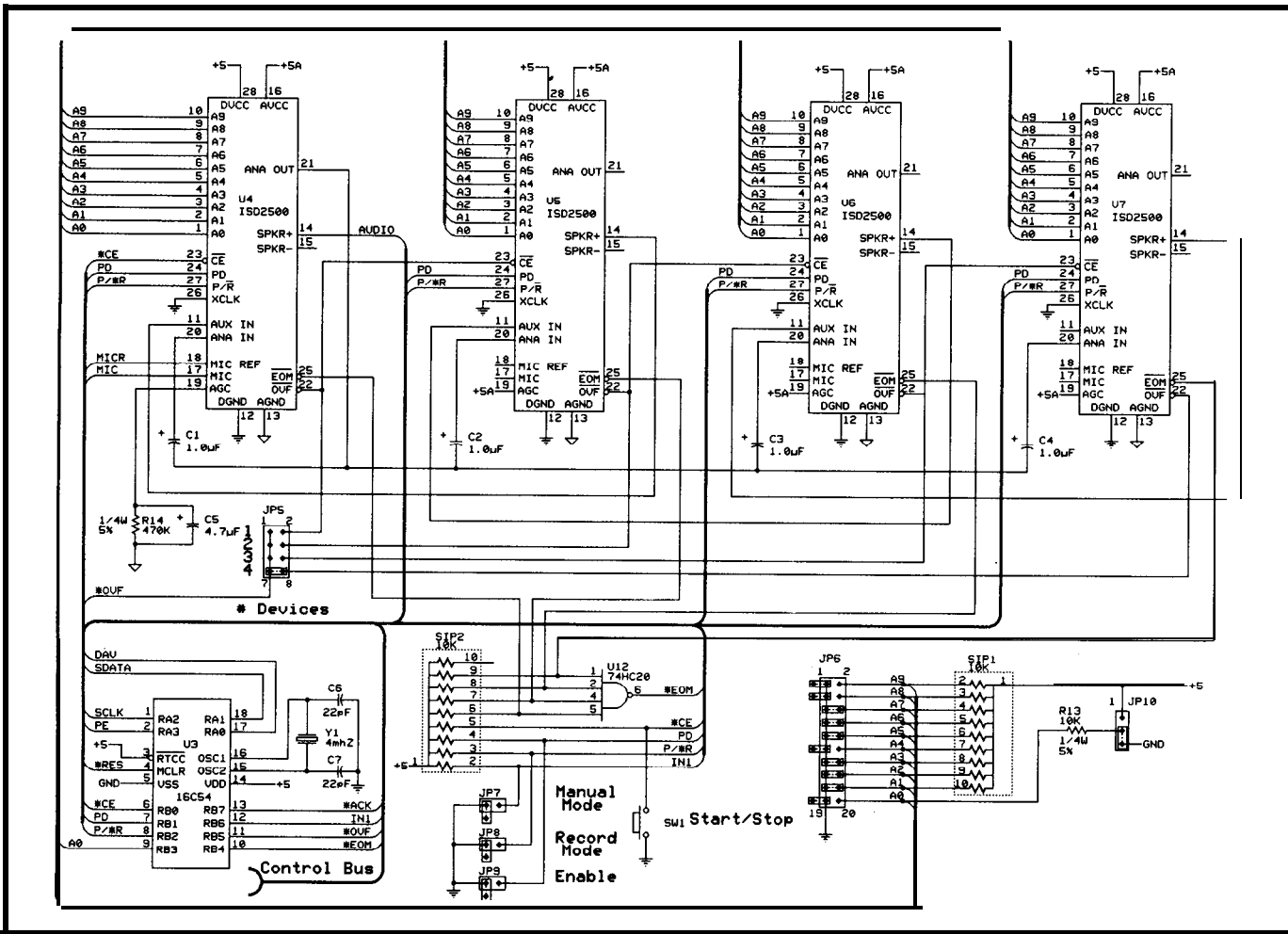


Figure 2b—At the core of the board is up to four ISD2500 parts capable of holding up to 6 minutes of audio. A PIC processor controls the show.

PRERECORDING

The Message Board can't speak unless you provide it with a message base. I suggest you start by writing down a list of all the messages (or sounds) you wish to include. Practice repeating them a few times until you feel comfortable. The Message Board can be put into manual record mode using the following steps:

Insert the "Manual Mode" jumper to instruct the micro to release all of the control lines to the ISD2500s. Insert the "Record Mode" jumper to place the ISD2500 devices into the record mode. Insert the "Enable" jumper to remove the ISD2500 devices from power-down mode.

Now plug an electret microphone into the audio input jack. You may begin recording the first message on your list by holding down the "Start/Stop" button until you are finished with the first message. Each time the button is released, an EOM marker is

inserted. You may continue to record additional messages until the OVF LED comes on. At this point the storage area is filled and will not accept additional input. If you wish to rerecord all the messages, remove and

then reinsert the "Enable" jumper. You are now ready to rerecord all of your messages. Whenever the record/playback mode is changed, the address pointer is reset within the ISD2500s, requiring you to start over if you wish

Listing 1—The Message Board is a memory-mapped device when plugged into an RTC52 and can be controlled directly from BASIC.

```

10 PRINT
20 INPUT "Phrase ii (1-255, 0=stop)" X
30 PRINT "Hit any key to cancel status reporting"
40 XBY(0E080H)=X: REM Write phrase number to port
50 Y = 255: REM Initialize to a nonstatus value
60 X = (XBY(0E090H).AND.7): REM Get status from port
70 G = GET: IF G<>0 THEN GOTO 10: REM Key pressed?
80 IF X=Y THEN GOTO 60: REM If old=new, skip status print
90 Y = X: REM Update old status
100 IF (X.AND.1)=1 THEN PRINT " Busy ",
110 IF (X.AND.1)=0 THEN PRINT " Not Busy ",
120 IF (X.AND.2)=2 THEN PRINT " Data Available ",
130 IF (X.AND.2)=0 THEN PRINT " Data Captured ",
140 IF (X.AND.4)=4 THEN PRINT " Acknowledged "
150 IF (X.AND.4)=0 THEN PRINT " Working "
160 IF X<>4 THEN 60: REM This value indicates end of phrase
170 GOTO 10

```

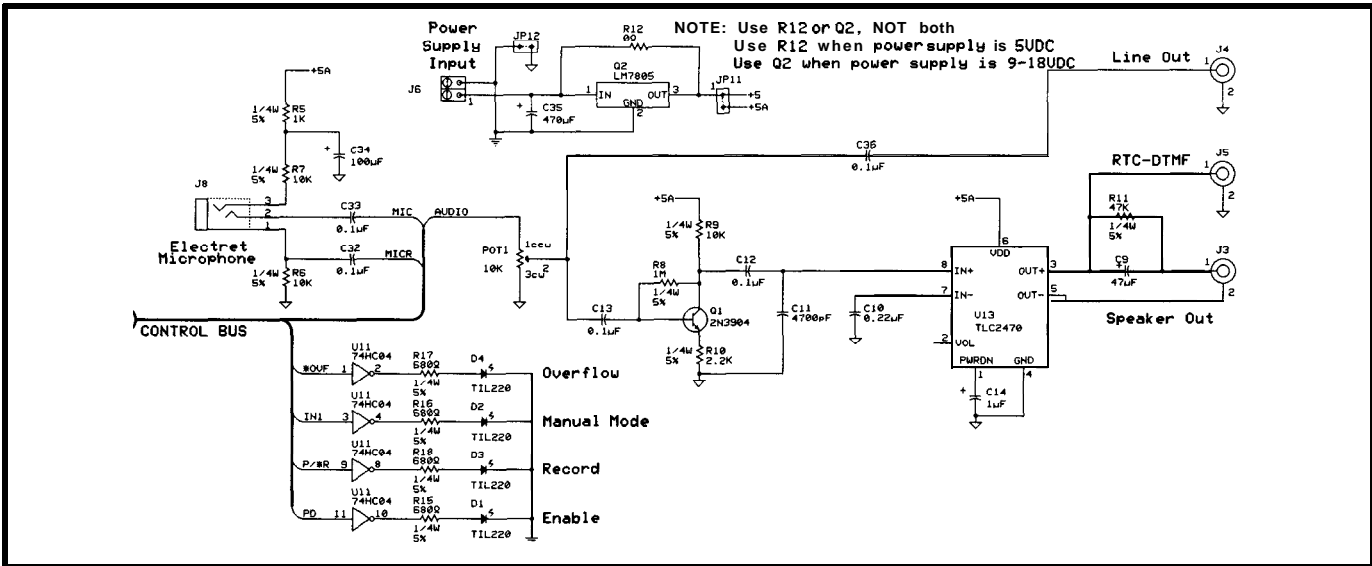



Figure 2c—An electret microphone is used to record audio. Even though the ISD2500 chips include amplifiers, one is included on the board so it functions the same as the HCS-Voice board.

to correct an error or add more messages.

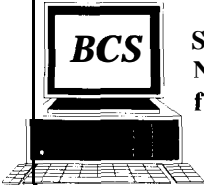
You may manually review your messages by removing both Record mode and Enable jumpers and reinserting only the Enable jumper (playback mode). Each time the Start/Stop button is pressed, a message will play.

When you are pleased with the session, remove the "Enable," "Record Mode," and "Manual Mode" jumpers. The micro has now regained control and you may command the Message Board to speak.

I have provided a couple of short no-frills programs to test out the

Message Board once you have filled its mind with message phrases. The program in Listing 1 uses the Message Board as a memory-mapped I/O device with an RTC52. Notice the "Parallel Port Interface" in the schematic is not necessary when the Message Board is to be used with either the RTC or HCS

EARN \$5000 PER MONTH With your home computer!



Spending too much on your computer? Now it's time to let it earn good money for you! Our two successful Software Packages, MoneyMaker Volume-1 and MoneyMaker Volume-2 will give you all the insider information you need to start and succeed with your own PROFITABLE BUSINESS Part-Time at home!

Your cost for all this extra income?

Only \$29.95 for MoneyMaker Volume-1

**For BOTH VOLUMES NOW send just \$39.95 !!!!
AND GET NEW 300 PAGE MANUAL FOR 1994**

**Includes FREE Talking Demo Disk and
3 BIG Catalogs full of more Money Making ideas!**

* We always pay postage *

Please specify disk size or we ship 3.5"

- Check, Credit Card, or M.O. to:

BUSINESS COMPUTER SYSTEMS, INC.

46-16 65TH PLACE -Dept M2

WOODSIDE, N.Y. 11377

Quality Software for the PC since 1986

AMEX, VISA, MC ORDERS
FAX (718) 898-3126
PHONE (718) 803-3638

#12

TIRED OF WAITING FOR THE PROMPT ?

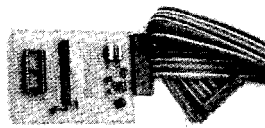
Speed up with a ROM DRIVE! Boots DOS and programs instantly. Also used to replace mechanical drive completely in controllers or diskless workstations. The only perfect protection from viruses. Easy to install half-size card.

VVDISK1 128k \$75
VVDISK2 720k \$150
VVDISK3 288m \$195

\$75

Quantity discounts!

DOS IN ROM!



\$95 EPROM PROGRAMMER

- Super Fast Programming
- Easier to use than others
- Does 2764/27080 (8 Meg)

WORLD'S SMALLEST PC !!!

ROBOTS ALARMS RECORDERS DOS

THREE EASY STEPS: \$27 1K QTY
1. Develop on PC
2. Download to SRC
3. Burn into EPROM \$95 SGL QTY

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

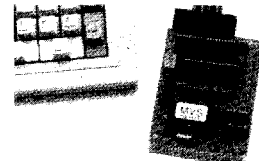
FREE SHIPPING IN U.S.

5 YEAR LIMITED WARRANTY

MVS

Box 850
Merrimack, NH
(508) 792 9507

8088 SINGLE BOARD COMPUTER



#12

Listing 2—The board may also be controlled from BASIC when plugged into a PC's printer port

```

10 REM Possible printer port addresses are &H278, &H378, or &H3BC
20 DPORT=&H378: CPORT=DPORT+2: SPORT=DPORT+1
30 PRINT: INPUT "Phrase # (1-255, 0=stop)",X
50 PRINT "Hit any key to quit status reporting"
60 OUT DPORT,X: REM Write phrase value to the data port
70 OUT CPORT,(INP(CPORT) AND 254): REM Strobe low
80 OUT CPORT,(INP(CPORT) OR 1): REM Strobe high
90 Y = 255: REM Initialize to an impossible status value
100 X = (INP(SPORT) AND &HE0): REM Read status port
110 IF (INKEY$<>"") THEN GOTO 30: REM Any key exits status routine
120 IF (X=Y) THEN GOTO 100: REM If old=new, skip status printing
130 Y = X: REM Update old status
140 IF (X AND 128)=128 THEN PRINT "Data Accepted ",
150 IF (X AND 128)=0 THEN PRINT "Data Available",
160 IF (X AND 64)=64 THEN PRINT "Data Acknowledged".
170 IF (X AND 64)=0 THEN PRINT "Processing Data",
180 IF (X AND 32)=32 THEN PRINT "Playback Busy"
190 IF (X AND 32)=0 THEN PRINT "Playback Not Busy"
200 IF (X=&HC0) THEN GOTO 30: REM Playback has finished
210 GOTO 100

```

system controllers. The program in Listing 2 uses GWBASIC to write directly to your PC's parallel port. The "RTC/HCS Interface" is not needed if you wish to use the Message Board stand-alone with a printer port.

Happy messaging. ☐

Jeff **Bachiochi** (pronounced "BAH-key-AH-key") is an electrical engineer on the *Computer Applications Journal's* engineering staff. His background includes product design and manufacturing. He may be reached at jeff.bachiochi@circellar.com.

SOURCE

Circuit Cellar Kits
4 Park St.
Vernon, CT 06066
(203) 875-2751

Message Board (full kit] with one 90-sec. storage device \$149.00

Message Board (assembled and tested) with one 90-sec. storage device \$199.00

Individual ISD chips are *not* available from Circuit Cellar (except with purchase of the Message Board).

CONTACT

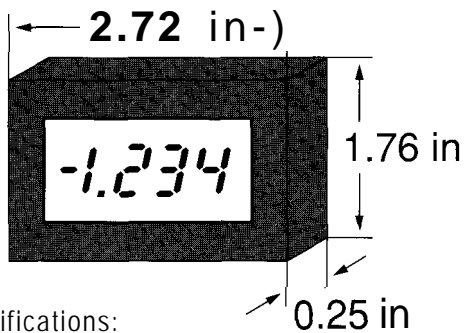
Information Storage Devices
2841 Junction Ave.
San Jose, CA 95134
Tel: (408) 428-1400
Fax: (408) 428-1422

IRS

413 Very Useful
414 Moderately Useful
415 Not Useful

3½-DIGIT LCD PANEL METER

-Available now at an unheard of price of **\$15** plus s & h
New! Not surplus!



Specifications:

- Maximum input: f199.9 mV
additional ranges provided through external resistor dividers
- Display: 3½-digit LCD, 0.5 in. figure height, jumper-selectable decimal point
- Conversion: Dual slope conversion, 2-3 readings per sec.
- Input Impedance: > 1 OOM ohm
- Power: 9-1 2 VDC @ 1 mA DC

Circuit Cellar, Inc.

4 Park Street, Suite 12, Vernon, CT 06066
Tel: (203) 875-2751 Fax: (203) 872-2204

0110
1111
0110

DATA

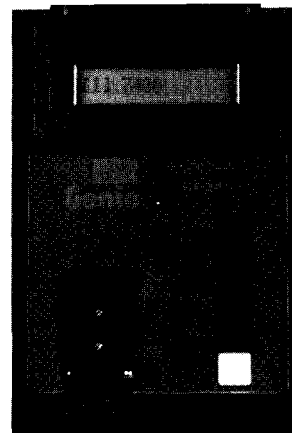
Genie™

PORTABLE DIGITAL IC TESTER

EXPRESS SERIES HT-28

The Data Genie HT-28 is a convenient way of testing Logic IC's and DRAM's Tests most TTL 74, CMOS 40/45 and DRAM's 414000, 44164-441 000 IC's.

The HT-28 can also identify Logic IC's when a non-standard part number is used or if the product ID has been removed. The HT-28 is compact in size, 6-5/16"x4-5/16"x 2-1/8", and can be operated from an AC adaptor or up to 600 hours on 4-"AA" alkaline batteries.



Dealer Inquires Welcome

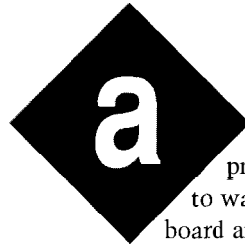
MING
Microsystems
DIVISION OF MING & P. INC

17921 Rowland St.
City of Industry, CA 91748
TEL (818) 912-7756 FAX: (818) 912-9598

When The SHARC Bites

SILICON UPDATE

Tom Cantrell



As spring approaches, it's time to wax up the long board and head over to

Santa Cruz to catch a few curls. If you're ever there, make sure you stop by the Surfing Museum at Lighthouse Point and check out the gallery of surfboards that paid the ultimate price as toothpicks for sharks.

Now, a surfboard is very hard, easily surviving—as I know first-hand—being run over by a car, falling off one at freeway speeds, hammering a tent stake, and so forth. The samples in the surf museum are striking, not being chewed, mangled, torn, or bent. Rather, the part now serving as roughage in some sharks weight-loss plan is simply gone—cleanly removed as with a scalpel.

Sometimes when I'm waiting for a wave, an inquisitive seal will pop up—how cute, eh! However, any feelings of affection are quickly tempered by the realization that when I'm in a black neoprene wetsuit, I look a lot like a well fattened seal-favored appetizers of our predatory friends. Makes me feel kind of like I've got a flashing neon "Eat At Tom's" sign on my forehead.

Circling the waters of the RISC versus DSP controversy is a JAWS (Just Another Wild Semiconductor) of a different sort: the 21060 SHARC from Analog Devices.

The debate that rages on is whether regular RISC CPU chips can handle signal processing chores as well as DSPs (see "Is the Am29050 a FIR-bearing Animal?" in the September '93 issue). Alternatively, the question could be whether DSP chips can handle regular processing as well as

RISCs? As in most religious wars it's a case of "where they stand is where they sit"; that is, the right answer (and question) correlates highly with what a particular outfit is trying to sell you.

Like a bullet manufacturer, you stand to benefit from this war, whomever wins. Both camps are tuning, tweaking, and packing transistors in a feverish effort to achieve at least a temporary advantage. Notably, most new (Not-So-) RISC chips are appearing with built-in MAC (Multiply & Accumulate) instructions to speed up "DSP operations."

So, is the SHARC, like its ocean going counterpart, the ultimate predator? Or, as the RISCers believe, is it just a big dummy with lots of teeth, as wimpy as our local hockey team of the same name? They're so bad, locals were cheering their quest for the record-of most consecutive losses!

IVY LEAGUE IC

SHARC, standing for "Super Harvard Architecture," joins ranks with Superscalar and Superpipeline (I just call 'em Superdupers) in the performance-at-any-price quest for king of the hill bragging rights.

Most of the microprocessors you're familiar with—such as the Z80, 'x86, 68k, etc.—rely on a single bus for both instruction and data delivery.

By contrast, a Harvard architecture uses two separate buses—one for instructions and one for data—allowing simultaneous instruction fetch and data read/write. For full parallel operation, each data bus has a corresponding address bus. These systems have four buses, which are often referred to as *PA*, *DA*, *PD*, and *DD* (program address, data address, program data, and data data, respectively).

The problem with lots of buses is they require lots of pins, the main reason Harvard chips are as rare as Great Whites. However, besides pin count, there are other concerns about the suitability of Harvard architectures for general-purpose processing.

For instance, if instruction and data buses are physically separate, normally trivial issues such as loading a program into memory or embedding



You say you need a hot topic to get your

mind off the weather? Picture yourself in a lawn chair, Hawaiian shirt, and shorts with a Long Island Ice Tea on the beach reading the latest on SHARCs. No, not the fish, the chip...

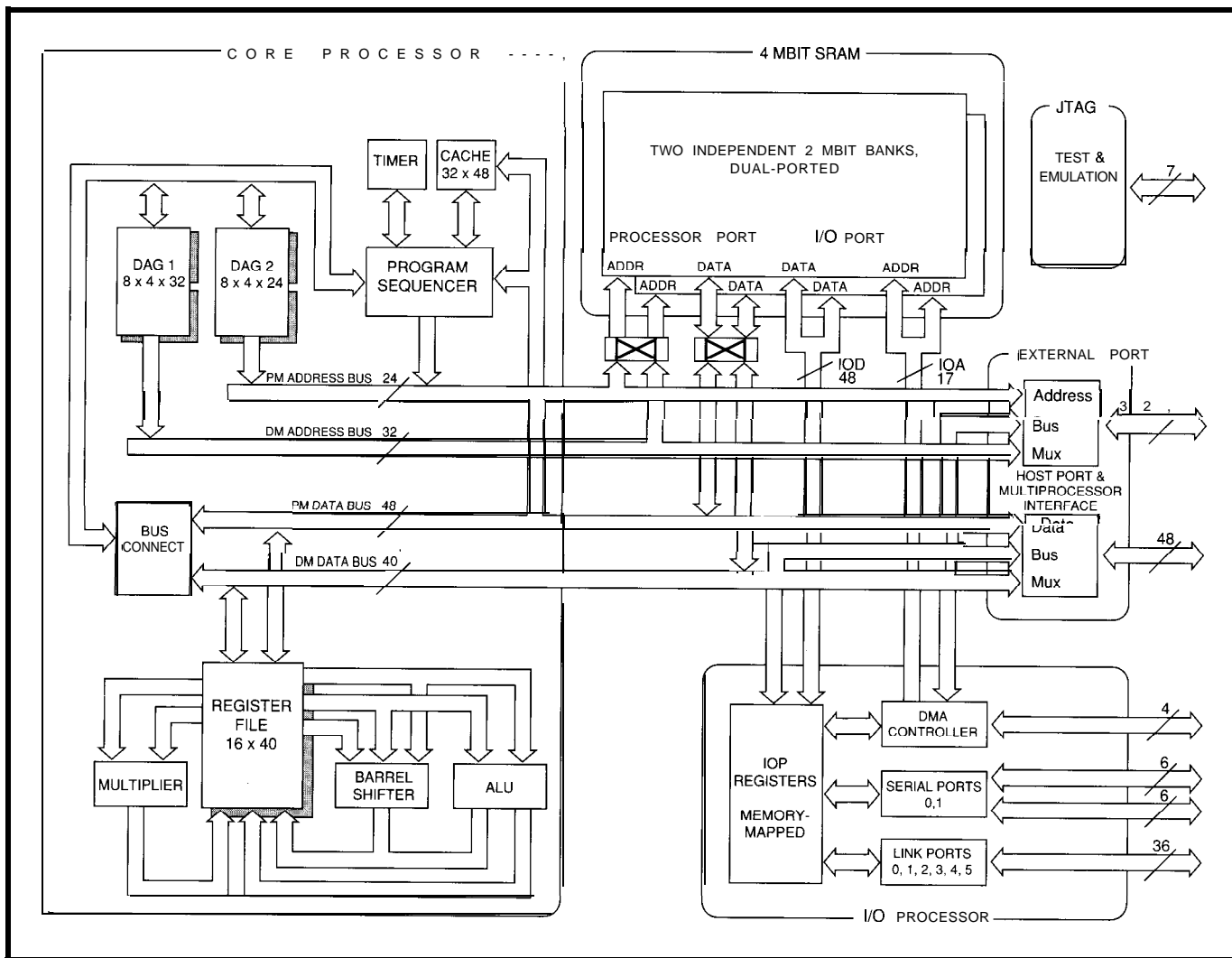


Figure #1 The SHARC (super Harvard Architecture) sports four separate buses (two for address and two for data) plus loads of on-chip memory.

constants in a program ROM may be problematic since both require accessing (the former writing, the latter reading) data in program memory. Typically, special instructions such as LOAD CONSTANT (i.e., do a load from the instruction memory) or bus override/swapping schemes are needed.

Another concern is making sure you get full "bang per buck" out of the bandwidth. Yes, performing an instruction and data access every cycle is twice as fast as doing them one after the other, but how likely is it that your program acts that way? Often instruction fetches and data accesses don't match 1:1, meaning all those pins you paid for aren't being fully utilized.

In a similar fashion, asymmetric access times can prove troublesome. What if instruction fetches are out of a

200-ns EPROM while data ops access a speedier 100-ns SRAM? Full utilization of all bus bandwidth requires that instruction and data access times be equal.

THROWN FOR A LOOP

Having populated the anti-Harvard army with straw men, now let's see how the SHARC (Figure 1 and Photo 1) chews them up.

The chip starts with the existing 21020, a high-end (40 MIPS) 32-bit floating-point DSP, whose Ivy League credentials are clearly established by the four buses previously described.

Inclusion of an instruction cache is the first feature that qualifies the SHARC for "superness." When instructions are available from cache, the PD bus is free to fetch data along with the DD bus. Thus, the SHARC one-ups a regular "non-super" Harvard

design by performing an instruction and two data ops at once.

Yes, the cache is quite small (32 instructions), but remember that the core of typical DSP routines is a tight multiply and accumulate loop that goes something like:

```
FOR I = 1 TO NUM_TAPS
  ACC = ACC + (DATA(I) *
    COEF(I))
NEXT I
```

By the way, notice that this tight loop is more "data bound" than typical non-DSP code, meaning there are relatively few instructions compared to data accesses instead of vice-versa. The instruction/data parallelism of DSP code is well served by the Harvard architecture (which helps explain why the AMD 29k, a Harvard design, performed so well in the article I

mentioned above). By offering two data ops per cycle, the SHARC is presumably well balanced for these tight loops-ideally, all three buses are active all the time.

By far the most exciting feature of the SHARC is the inclusion of a whopping 4 megabits of SRAM. Considering that 4-megabit SRAM chips are themselves barely peeking out of the lab, it wouldn't be unreasonable to consider the SHARC a smart memory chip rather than a DSP. Rather a case of the tail wagging the shark, wouldn't you agree!

While it is outrageous, packing so much memory on-chip yields equally dramatic benefits. The SRAM is very fast, with the 15-ns or so access time needed to meet the zero-wait-state requirement of the 25-ns (40 MHz) cycle time. Despite the claims of zealots, in my opinion performance is

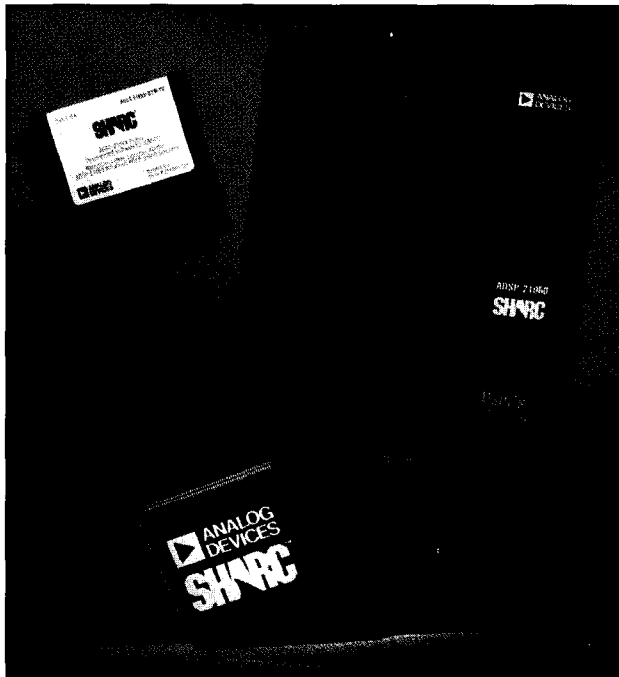


Photo 1-The 21060 SHARC from Analog Devices has stirred up a debate between RISC chips and DSPs and which of these handles signal processing better.

mainly related to memory bandwidth rather than instruction set complexity or other architectural embellishments. The SHARC, shipping up to 16 bytes

(six from I-cache and five each on the PD and DD buses) at 40 MHz yields a whopping 640 megabytes per second of bus bandwidth. This kind of bandwidth would be very hard [and expensive] to achieve if the memory-and all those bus lines-had to go off-chip.

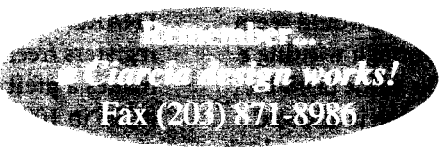
Hey, with more buses than a Greyhound terminal, why not go for the gusto? The SHARC adds an I/O processor with yet another bus. The IOP contains a lo-channel DMA controller, clocked serial ports (typically used to connect the A/D and D/A subsystems) and link ports which can connect up to six SHARCs—at a speedy 240 MB/second-in a multiprocessor configuration.

THANKS FOR THE MEMORIES

While providing speed and saving pins, the on-chip memories are cleverly organized to overcome some

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. We have a team of accomplished programmers and engineers ready to design products or solve tricky engineering problems. Whether you need an on-line solution for a unique problem, a product for a startup venture, or just experienced consulting, the Ciarcia Design Works is ready to work with you. Just fax me your problem and we'll be in touch.



"Is the
TEARROR
 of **BIG S-L-O-W** software
 stalking your job security?"

Take some of the bytes out of your bloated code!
 Call Franklin Software, the first and last
 word in quality for
 your 8051 and 80C166 software projects!

**F. FRANKLIN
 SOFTWARE, INC**

Call for your free Release VI Evaluation Kit!
 tel: (408) 296-8051 fax: (408) 296-8061

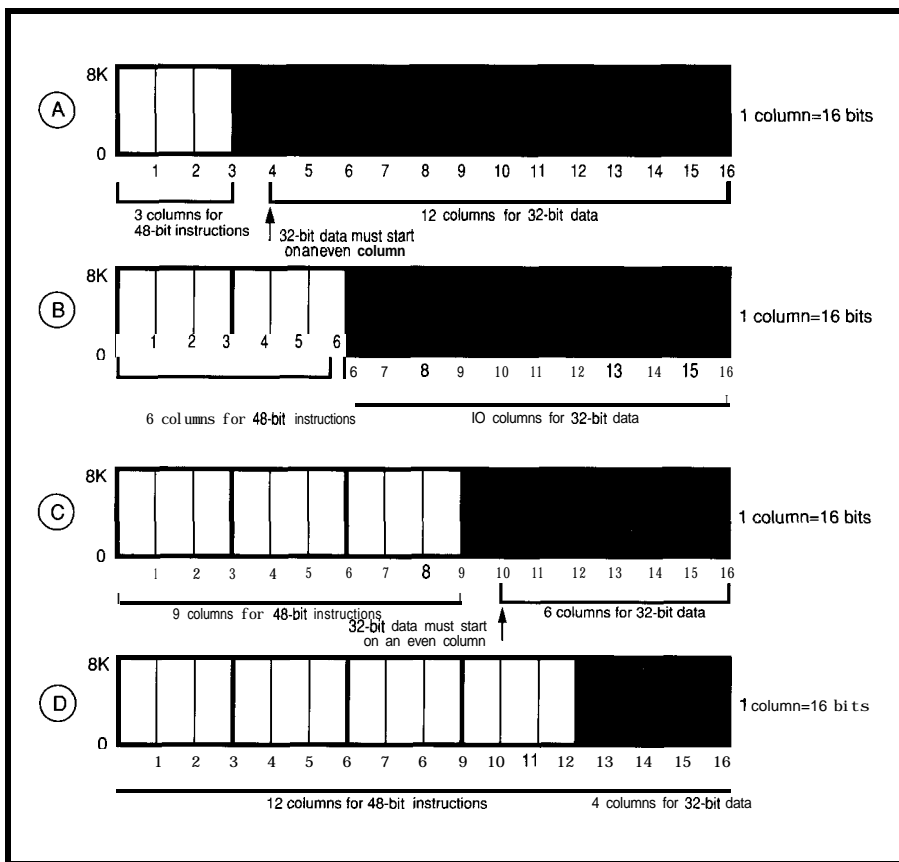


Figure 2--Memory configurations for the SHARC.

of the "gotchas" associated with Harvard designs.

For instance, the previously mentioned program load/constant problem (i.e., accessing instruction memory as data) is handled by cross-bars so that either program or data memory may be accessed via either bus. Indeed, both can be accessed from one of the buses, but the non-Harvard penalty (i.e., accesses are back-to-back rather than in parallel) is automatically incurred. Wise programmers will allocate memory to fully utilize all bus bandwidth. For instance, in the previous code fragment the instructions should be in cache, the coefficients (which are constants during the life of the loop) in instruction memory, and the data being worked on (i.e., the signal) in data memory.

Given that code and data can both reside in either of the two independent 2-megabit blocks, a basic concern is how the allocation can be divided. A simple-minded scheme might just split it in half, or otherwise arbitrarily define it according to the chip designer's whim. Naturally, whatever they

choose would be just slightly off from your application's needs leading to headaches and wasted RAM.

All this switching and allocating is complicated by the fact that the instruction size is 48 bits while data types include 16-bit integers and 32-bit and 40-bit floating point.

The chip goes to great lengths to make packing and moving stuff around easy. Each 2-megabit block of RAM is physically organized as 16 columns, each 16 bits, in 8K rows. When mixing, for example, 48-bit instructions and 32-bit data in the same bank, the instructions call for three contiguous columns while the data uses two. Figure 2 shows the possible combinations of 48-bit instructions and 32-bit data. Despite AD's best efforts, notice the loss of a column in cases A and C.

Similarly, mixing 40-bit and 32-bit data in the same block would seem dicey. However, through wasting a byte for each 40-bit value (a 48-bit access is used), mixing is allowed thanks to bus switching logic which routes the access via the instruction bus.

Mixing 16-bit and 32-bit data is naturally more straightforward. A RAM block is actually allocated two memory spaces, so the type of transfer (16- or 32-bit) is simply determined by which space is accessed.

All this shuffling, switching, and sizing may sound kind of grim, but actually AD has done an admirable job of dealing with the inevitable granularity and access restrictions posed by the Harvard scheme. Indeed, a little tricky programming—such as dynamically changing the RAM configuration bits—can even get around some of these, though the data sheet warns that an "in-depth" understanding is required to stay out of trouble.

MILD-MANNERED MAINFRAME

Fittingly, this Superchip of steel is surprisingly mild-mannered on the surface. Since so many buses—and all their messy control logic—are on-chip, the external interface is refreshingly straightforward (Figure 3).

All the Harvardness is kept on chip, so to the outside world the SHARC looks like a regular single data bus system. The address bus (32 bits) and data bus (48 bits) access the external world in a conventional manner using RD* and WR* pins. Four MS* (memory select) pins are provided to enable external memory or I/O devices. If necessary, ACK acts as a ready input to inject wait states.

However, an on-chip wait-state generator does an excellent job, even providing for selectable precycle (to avoid bus contention) and/or postcycle (to meet address/data hold time) waits.

Though the chip doesn't directly drive DRAMs, it does give an external DRAM controller the hooks to support page-mode (PAGE pin) and synchronous (ADRCLK pin) devices.

For host processor access, the bus "turns around" and becomes a slave to the CS* input. When CS* is asserted, an external CPU can access the SHARC's internal memory and I/O registers freely. Indeed, the SHARC makes a pretty nifty, though expensive (\$296 in 1000s), 128Kx32 SRAM.

Similarly, on-chip resources can be accessed via the six "link" ports—synchronous nybble-wide ports that

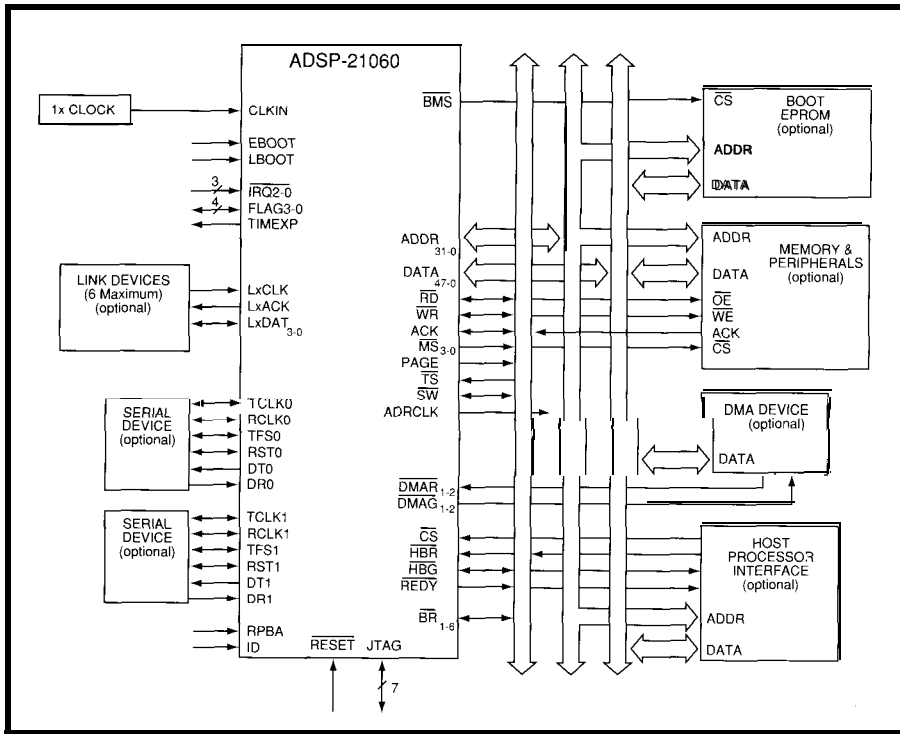


Figure 3—Since the SHARC has so many of its buses on the chip, external interfacing is much easier than it could be.

serve as the basis for ganging SHARCs up on a stubborn problem. Portions of each SHARC's address space are mapped into a "global" memory, in essence combining the resources of multi-SHARCs into one logical machine.

The JTAG (Joint Test Access Group) port is appearing on more and more micros. It's kind of a clocked serial port that gives access to the device internals for testing and debugging. JTAG is the hook used by the EZ-ICE emulator which combines a Windows-based front end with most of the hardware control (except real-time trace) of a traditional (and more expensive) ICE.

Based on SRAM, the SHARC needs a way to clear the cobwebs on power-up. Figure 3 shows how a simple x8 EPROM can do the initial dirty work—a more pleasant prospect than the six boot ROMs the SHARC's 48-bit instructions would otherwise demand.

However, AD went further by anticipating the in-system programmable trend I covered a few issues back. So, the SHARC can also be "booted" from a host processor or even via the link and JTAG ports. Oh yeah, it can also boot from external memory,

so if you want six boot ROMs, you got 'em.

BENCHMARKBROUHAHA

For development, AD offers a GNU-based C compiler package including assembler, simulator, and so forth for a reasonable \$995. In fact, AD has been heavily involved with something called NCEG: the Numeric C Extensions Group. The extensions, which support DSP applications with array operations, looping, circular (buffer) pointers and so on, have been incorporated in the latest release (2.4) of the GNU C compiler.

A Numeric C version of the previous example looks something like:

```
iter i= num_taps;
acc = sum (coef [i] *
           data[i]);
```

Notice the `i t e r` loop designator and the `sum` array operator—shades of APL!

You may argue it doesn't make the coding a heck of a lot better, and you're right. Actually, these extensions are really designed to help the compiler generate optimized code for machines (like DSPs) that support

looping, arrays, and so forth in hardware.

AD's own benchmarks claim the SHARC is 2–4× faster than competing DSPs. The only benchmark versus RISC is a 1k complex FFT against the i860 which the SHARC performs 40% faster (0.46 ms vs. the RISC's 0.76 ms).

But the article I cited above showed the AMD 29050 was 15 % faster than the '860 (2.78 μs vs. 3.3 us) for a 95-tap FIR filter. Interestingly, AD's own benchmarks include a 95-tap FIR filter as well at 2.5 us.

Now, I certainly don't have complete benchmark information, so this could all be apples versus oranges, and besides, everyone knows benchmarks can be used to promote many versions of the truth. Nevertheless, I'm getting the feeling that, while the SHARC may be on top today, the RISC versus DSP battle is far from over.

BLOOD IN THE WATER

Did you know that sharks don't have the adjustable bladders lesser fish use for flotation. Rather, the needed buoyant gases are generated as a byproduct of digestion. The hungrier a shark gets, the harder it has to work to stay afloat.

Get the point? Whichever side—RISC or DSP—you're on, it's clear the high-stakes battle for micro supremacy is, as for a shark, an eat or die proposition. 🦈

Tom Cantrell has been an engineer in Silicon Valley for more than ten years working on chip, board and systems design and marketing. He can be reached at (510) 657-0264 or by fax at (510) 657-5441.

CONTACT

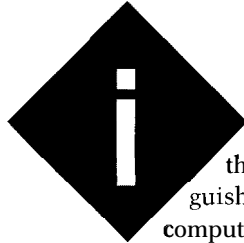
Analog Devices, Inc.
One Technology Way
Norwood, MA 26062
(617) 461-3881
Fax: (617) 821-4273

416 Very Useful
417 Moderately Useful
418 Not Useful

Who's in Control?

EMBEDDED TECHNIQUES

John Dybowski



It could be said that what distinguishes embedded computers from their

office-bound counterparts is the fact that the embedded variety are primarily designed for unsupervised operation. These things are forever finding their way into potentially troublesome situations where they are either dispersed about remote locations or placed into seemingly inaccessible predicaments. In any case, it seems wherever they end up the electrical and environmental conditions are usually less than favorable. In spite of these difficulties, we assume these devices will operate for extended periods without the need for human intervention.

Many different applications can be served, but the most demanding casts the embedded computer in the role of a system controller. Should such a control-oriented system malfunction, the results can range from mildly irritating to utterly destructive; there's nothing worse than a system controller running out of control. In the majority of cases, this loss of control can be attributed to electrically hostile conditions, and some external event can be blamed for the system malfunction. However, it pays to keep an open mind since the culprit may be indicative of some design inadequacy.

One of the worst offenders is the typical RC processor reset circuit. You will, no doubt, experience no trouble using such a circuit throughout the course of your product development. But chances are good that once you've moved your system away from the lab bench, you may begin to experience

new problems. Remember: it's an embedded controller's lot in life to be around equipment that wreaks havoc with the power grid. In such a setting, it's not uncommon to find out that your controller checks out every time some big motor kicks in. Apparently what's happening here is that the power dips to a point where loss of regulation occurs. Once V_{cc} drops out of spec, all bets are off. Unfortunately, a simple RC reset circuit that works so well on the lab bench is sufficiently primitive to be of no use at all in such a situation. Obviously, one way to solve this problem is to incorporate a decent reset circuit into the design...but it pays to look further.

Once you start examining all the real-world ills that can trouble your apparatus, it becomes obvious that the construction of a truly bullet-proof system can be extremely difficult. Fortunately, although most embedded computers operate under electrically hostile conditions, only a small percentage of these have to be truly impervious to any kind of interruption. Most systems are amenable to temporary disruptions as long as control can eventually be restored. Clearly, a single-chip implementation that only knows how to cold boot has quite different needs than a complex data collection system that is charged with collecting and retaining megabytes of data over extended periods of time.

Assuming the embedded computer is designed properly to begin with, some desirable attributes can be identified that can strengthen robust operation. These include a V_{cc} referenced reset circuit, a power-fail indicator/interrupt, and a watchdog to reset all subsystems if all else fails. Having mentioned that the DS2250 possesses the capability to operate under adverse conditions, it shouldn't be surprising to find that these features all come built-in.

DAMAGE CONTROL

The DS2250 generates an internal power-on reset without the need for external components. When V_{cc} is below the reset threshold, the DS2250 is held in an internal reset state. Once



In his continuing series on building an embedded control system, John looks at reset circuits, power fail conditions, more battery management, and finally expands on his treatment of the LCD display interface.

POWER CONTROL REGISTER

Label: PCON Register Address: 087H

D7	D6	D5	D4	D3	D2	D1	D0
SMOD	POR	PFW	WTR	EPFW	EWT	STOP	IDL

PCON.6: POR

Power On Reset: Previous reset initiated during power on sequence.
 Initialization: Cleared to 0 when power on reset occurs. Remains at 0 until set to 1 by software.
 Read Access: Can be read normally at any time.
 Write Access: Can be written only by using Timed Access Register.

PCON.5: PFW

Pwr Fail Warning: Indicates potential power failure in progress
 Initialization: Cleared to a 0 on power on reset.
 Read Access: Can be read normally at any time.
 Write Access: Not writable.

PCON.4: WTR

Watchdog Timer Reset: Set to 1 following Watchdog Timer timeout. Cleared to 0 after read of PCON reg.
 Initialization: Set to 1 after Watchdog Timer Reset. Clear to 0 on power on reset.
 Read Access: May be read normally at any time.
 Write Access: Cannot be written.

PCON.3: EPFW

Enable Pwr Fail Interrupt: Used to enable or disable Power Fail Interrupt.
 Initialization: Cleared to a 0 on any type of reset.
 Read Access: May be read normally at any time.
 Write Access: Can be written normally at any time.

PCON.2: EWT

Enable Watchdog Timer: Used to enable or disable Watchdog Timeout Reset.
 Initialization: Cleared to 0 on No-V_{II} power on reset. Unchanged during other types of reset.
 Read Access: May be read normally at anytime.
 Write Access: Can be written only by using Timed Access Register.

PCON.1: STOP

stop: Used to invoke the Stop mode.
 Initialization: Cleared to 0 on any type of reset.
 Read Access: Can be read at anytime.
 Write Access: Can be written only by using Timed Access Register.

Figure 1a--The DS2250's power control register gives the user access to all kinds of features not found on a normal 8031 for dealing with reset and power fail conditions.

the voltage rises into the operating range, the internal reset generator counts a delay interval before releasing the processor for operation. This delay is established by counting 21,504 oscillator clock periods, which amounts to about 1.95 ms when running at 11.0592 MHz. This time does not include the period that V_{cc} takes to slew up to a valid operational level and the time it takes the crystal's mass to get into vibrational motion.

It pays to keep the reset interval as short as possible when operating from battery power, especially when the system is powered intermittently. If I'm trying to choke back the power consumption by only powering up the system to take a sample, it makes little sense to spend 100 ms or more waiting for the reset to let go. This isn't a consideration in most "normal" embedded applications. As evidence, many of the reset controller ICs on the market hold a very loose tolerance

on the reset duration, and in most cases will keep reset active for an excess of 1 second. In any event, these circuits work perfectly well in most fixed base applications, but just aren't appropriate for the type of system I'm developing here. Note that the DS2250's external reset pin is completely functional and can be used to place the DS2250 into reset at any time, but is unnecessary for implementing a normal power-up reset.

Most designs will use the reset pin in conjunction with $\overline{\text{PSEN}}$ to force the

DS2250 into bootstrap mode, which I'll be covering in a future column.

A built-in watchdog timer is provided as a means of restoring normal operation if the processor gets lost. The watchdog timeout period is equal to 122,880 machine cycles, which comes to about 133 ms at 11.0592 MHz. The nice thing about the DS2250's watchdog, unlike most outboard watchdog circuits, is that it can be turned on and off under firmware control. Personally, I wouldn't be without a watchdog, but depending on who's writing the code and what the code is doing, in some cases a watchdog could actually do more harm than good. In any event, it's useful to be able to keep the watchdog out of the way during initial firmware development.

All watchdog accesses must be performed using timed access, which is a method of protecting critical DS2250 bits and bytes from inadvertent modification.

INTERRUPT PRIORITY REGISTER

Label: IP Register Address: 0B8H

D7	D6	D5	D4	D3	D2	D1	D0
RWT		+	-	PS	PT1	PX1	PTO
							PX0

IP.7: RWT

Reset Watchdog Timer: When set to 1, Watchdog Timer count reset. Writing 0 to bit has no effect.
 Initialization: Cleared to 0 on any reset.
 Read Access: Cannot be read.
 Write Access: Can be written only by using Timed Access Register.

Figure 1 b--An extra bit is added to the interrupt priority register to give special consideration to the DS2250's watchdog.

Certainly the system watchdog qualifies as a critical system resource. Timed access is accomplished by writing two bytes to the timed access register located at C7h. The first write must be the value AAh with the second being a 55h. After this sequence, the protected region may be accessed, but there's a limit on how long this access window remains open. When the initial AAh is written, two timers are initiated. The first timer allows two instruction cycles until the 55h is written. The second timer restricts access to the protected area to within four instruction cycles of the first write (the AAh). Note that since this timer is started before the writing of the 55H, the time remaining for the actual access depends on the type of instruction that was used to write the 55h. If a one-cycle instruction was used to write the 55h, then three cycles remain to modify the protected bits. If a two-cycle instruction was used to write the 55h, then there are only two cycles left.

The enable-watchdog-timer bit is located at PCON.2 and is referred to as EWT. Although this bit requires timed access to write, it can be read normally at any time. The write-only watchdog reset bit is at IP.7. Figure 1 shows how these new bit assignments are squeezed into the PCON and IP registers. Listing 1 illustrates how timed access is used to enable and reset the watchdog function.

It's often a good idea to have a power fail interrupt so you can take care of any loose ends during a power failure before the micro fades to black. Incorporating this capability can be a problem with the limited interrupt resources available on a standard 8031. You seldom have enough interrupts to begin with, and the two-level priority structure and global enable makes things even more difficult. Of course, even if you do have a loose interrupt available, you still have to construct a power monitoring comparator.

The DS2250 has the power monitoring circuitry built in and adds a new power fail warning (PFW) interrupt to the standard 8031 interrupt set that overcomes prior limitations. The PFW interrupt is enabled by

Listing 1--Writing to the watchdog must be done within time limits set by the time access register.

```

TA      EQU      0C7H          ;TIMED ACCESS REGISTER
RWT     EOU      IP.7         ;WATCHDOG RESET
EWT     EOU      00000100B    ;ENABLE WATCHDOG TIMER

      MOV      TA,#0AAH
      MOV      TA,#55H        ;ENABLE TIMED ACCESS
      ORL      PCON,#EWT      ;ENABLE WATCHDOG

      MOV      TA,#0AAH
      MOV      TA,#55H        ;ENABLE TIMED ACCESS
      SETB     RWT            ;RESET WATCHDOG

```

setting the EPFW bit at PCON.3 to a 1. The corresponding power-fail warning flag is at PCON.5. Whenever V_{cc} drops below the low-voltage threshold, the PFW bit is set to a 1 and remains set until it is read by firmware. If the PFW interrupt is enabled, the processor will vector to location 2Bh on recognition of the low-voltage event.

To make this feature even more useful, the PFW interrupt is not controlled by the EA global interrupt enable bit. It can only be enabled or disabled by using EPFW, sort of a maskable nonmaskable interrupt. Furthermore, the PFW interrupt has the highest priority if it is enabled. The other interrupts can be programmed to either low or high priority using the standard two-level priority structure, but the PFW interrupt will interrupt any ISR if it is enabled.

One of the problems you pick up as a result of adding these recovery features is you must now make some conscious decisions on how to proceed if a problem occurs. Once you've taken steps to incorporating fault detection (and presumably tolerance) into your system, you've essentially relinquished the prerogative to be happy and dumb.

In the case of a reset event, the DS2250 lets you determine where you're coming from (kind of) so you can best approximate where you should be. This information is contained in some previously unused bits of the PCON special function register. Referring again to Figure 1, you can see that status bits exist which indicate that the most recent reset was a power-on reset or a watchdog timer reset. Additionally, if you are using the

power-fail interrupt capability, you can add your own indicators to signal a power failure incident.

Beyond just denoting the occurrence, the use of the power fail interrupt gives you capabilities that would not be available otherwise. Having an early warning of an impending power failure allows you not only to recover from an untimely loss of power after the event, but gives you a chance to orchestrate a organized, if not hurried, shutdown. But again, you must first decide what actually constitutes an orderly shutdown.

At one extreme you can enter into a benign code loop that tries not to do any harm while waiting for the power to either die or to return to normal. At the other extreme you might go so far as saving the processor's entire machine status in order to resume exactly where you left off on restoration of power. At first this sounds just like the thing, but in most cases the results prove to be less than desirable. In most real-time applications, there's a lot of setting up involved prior to actually processing any event. Then again, it's a pretty sure bet that the event of interest itself had passed into history long before operation was restored. Ironically, the type of processing that lends itself best to this type of resumption of operation is that which centers around performing computations and manipulations on stored data. This type of batch processing is not really the most common activity for an embedded computer.

As an alternative, you might want to record the system's operating state at the time of the power failure and then sort things out on power up. In

Listing 2—Miscellaneous LCD support services include the *putchar* function needed for driving the I²C-based 4 x 20 LCD panel.

```
#pragma LARGE_CODE
#include "REG5000.h"

/* Defined bits for LCD control over the I2C bus */
#define Lcd0 0x42
#define Lcd1 0x44
#define LCDs 2
#define DEN 0x20
#define DRS 0x10

extern void InitLcd();
extern void ClearLcd();
extern void PositionCursor(unsigned char c);
extern void LCD_OUT(unsigned char c);

unsigned char Cursor[LCDs];
unsigned char ActiveLcd;

/* Puchar to LCD */
void putchar(unsigned char c)
{
    extern void DataWr(unsigned char c);
    unsigned char b;

    if (c == '\n'){
        b = Cursor[ActiveLcd];
        if (b < 20)
            PositionCursor(20);
        else if (b < 40)
            PositionCursor(40);
        else if (b < 60)
            PositionCursor(60);
        else
            PositionCursor(0);
    }
    DataWr(c);
    return;
}

/* Position LCD cursor */
void PositionCursor(unsigned char c)
{
    extern void CommandWr(unsigned char c);

    Cursor[ActiveLcd] = c;
    if (c > 60)
        CommandWr((c 60) + 84 + 0x80);
    else if (c > 40)
        CommandWr((c 40) + 20 + 0x80);
    else if (c > 20)
        CommandWr((c 20) + 64 + 0x80);
    else
        CommandWr(c + 0x80);
    return;
}

/* Clear LCD and home cursor */
void ClearLcd(void)
{
    Cursor[ActiveLcd] = 0;
    CommandWr(1);
    return;
}

/* Write to LCD data register & handle cursor positioning */
static void DataWr(unsigned char c)
```

(continued)

this case, the operational state could be resumed or abandoned as appropriate. Naturally, there are as many approaches to fault recovery in embedded systems as there are embedded systems. Alas, this is one area that must often be reinvented to precisely suit the particular system.

LCD CONTROL

Last month I showed you some code for driving the ec.25's K-based keyboard. Space limitations prevented me from finishing up with the display side of things, so now I'll wrap things up with a discussion of the LCD firmware support package. As with the keyboard driver, you will find all of the low-level functions required to handle a standard character-based LCD panel along with a way to hook into some standard C library functions. Since we picked up an ec.25-compatible get key replacement last month, this time around we'll acquire a new *putchar*.

At the heart of just about every character-based LCD is the HD44780 LCD controller. And although it does have some idiosyncrasies, it is by any estimation an unbelievably successful design. To those who have employed small LCDs in their designs, the problems associated with driving these things are already understood. In any event, much has already been written on the subject, so I promise to be brief.

The fact that most LCDs operate under control of the HD44780 is, at the same time, both an advantage and a liability. It's an advantage, because once you've figured out how to use it, you're all set to handle just about any character-based LCD on the market. It's a liability since because it must be coerced to drive so many different display configurations, it has a multitude of options. Because of this, it is, out of necessity, somewhat quirky. Although there are a number of nagging problems associated with using the HD44780, perhaps the biggest turns out to be the lack of correspondence between physical and logical cursor positions. In actuality, this is not so much an LSI problem as an implementation problem, but nonetheless it is an affliction that is unique to LCDs.

I'll now describe some support functions that I've coded up using C. Although there isn't a one-to-one correspondence, I've already presented some assembler routines in a previous column that perform similar functions to those I'll show here. After all, how many different things can you do to an LCD, anyway? If you're interested in seeing how they compare against this month's offering, check out my column in issue 35, "Putting I²C Through Its Paces."

Something to keep in mind as you follow Listing 2 is that the code is written to support multiple LCD panels over the same two-wire bus. This capability only results in a slight increase in code complexity.

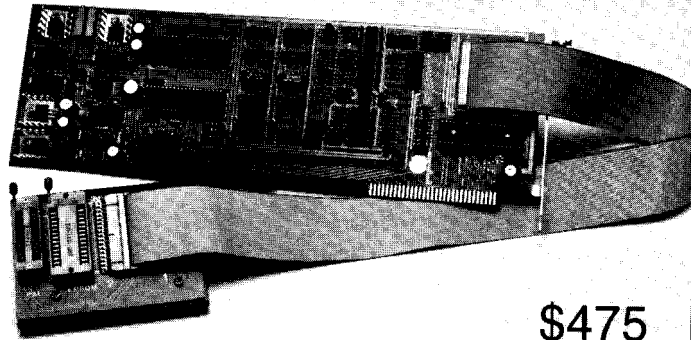
At the bottom of the pile is LCD-OUT. This routine merely passes a couple of arguments from the C code to the assembly level I²C driver. Since I support multiple LCDs and since, as I²C peripherals, they are addressable, this stub routine must determine the appropriate I²C address to pass to the driver routine. The present code supports two LCDs, which is enough to prove a point, but in principle the number could be much higher. You might wonder what I could possibly want with a bunch of LCDs on such a controller. In fact, it turns out I'm already at work on a modular facility surveillance/monitoring system that uses an independent LCD for each zone. As zones are added, a knockout is punched out on the control panel and an indicator display is added.

Since these low-level routines will ultimately be pressed to serve higher-level stream functions such as printf, I had to come up with a way to select the active display device on the fly without too much disruption. This is done by SelectLcd which accepts as its argument a number that denotes which display will become the active LCD and will respond to subsequent function calls. This number is stored and is made globally available for use by the various support routines that need to know. The active display remains live until a different display is selected.

With intelligent peripherals such as LCDs, the initialization sequence is

Universal Device Programmer

PAL
GAL
EPROM
EEPROM
FLASH
MICRO
87C51
PIC
93C46
XC1 736
PSD 3xx
5ns PALs

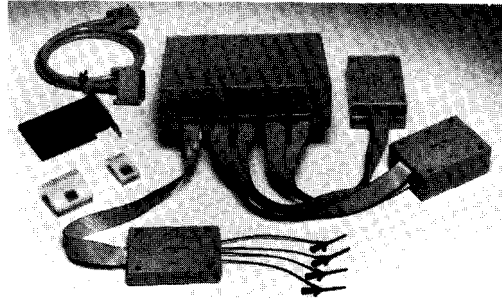


\$475

Free software updates on BBS
Powerful menu driven software

400 MHz Logic Analyzer

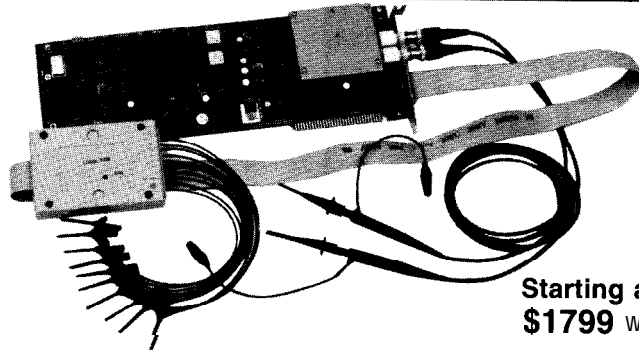
- up to 128 Channels
- up to 400 MHz
- up to 16K Samples/Channel
- Variable Threshold Levels
- 8 External Clocks
- 16 Level Triggering
- Pattern Generator Option



\$799 - LA12100 (100 MHz, 24 Ch)
\$1299 - LA32200 (200 MHz, 32 Ch)
\$1899 - LA32400 (400 MHz, 32 Ch)
\$2750 - **LA64400** (400 MHz, 64 Ch)

Price is Complete
Pods and Software
included

200 MSa/s DIGITAL OSCILLOSCOPE



Starting at
\$1799 With Pods
& Software

- 200 MSa/s sampling Rate
- 2 Analog Channels (2 ch. Digital Oscilloscope)
- 8 Digital Channels (8 ch. Logic Analyzer)
- 125 MHz Single Shot Bandwidth
- up to 128K Samples/Channel

Call (201) 808-8990



Link Instruments

369 Passaic Ave, Suite 100, Fairfield, NJ 07004 fax: 808-8786

undoubtedly the most important step you will perform. This is especially true in this particular implementation where I am using the display in nybble mode. The function `InitLcd` performs the necessary but admittedly arid steps of putting the LCD into a known state, configuring it for four-bit operation, and finally setting up the specific operational parameters. Incidentally, this is, in all respects, the same initialization routine that I described in issue 35. If you want to know what each instruction step actually does to the LCD, refer back to that column.

Since the LCD is operated as a write-only device, the current cursor address is maintained in global RAM. This information is not only needed for keeping track of the cursor for normal cursor positioning operations, but must also be used to preserve the apparent sequential nature of data as it is written to the LCD. If you look at `DataWr`, which handles the transfer of all displayable characters to the LCD, you can see how the cursor must be repositioned at key points where display continuity would otherwise be lost. On entry, the present cursor address is copied to local storage and a `switch` is used to catch addresses 20, 40, and 60. If the cursor is at any of these addresses, the LCD needs fixing so display data appears where it is expected. Additionally, address 80 is handled in a manner in which the LCD wraps back around to address 0. Finally, before the data character is transferred to the LCD, the current cursor address is incremented.

Now, in preparation for the dual nybble transfers, the data byte is dismembered and positioned. Control bits are appended before the byte is passed to the LCD-OUT routine. Since, for displayable data, the destination is the data register, the RS bit is raised to indicate the appropriate target register. Two transfers must be performed for each nybble since the enable (E) signal must be pulsed for each transfer into the LCD. Commands destined for the command register use the `CommandWr` routine, which simply performs the byte unpacking and transfer functions without any of the gyrations `DataWr` has to do. In this case, the command

Listing Z-continued

```

    unsigned char b;

    b = Cursor[ActiveLcd];
    switch(b) {
    case 20 :
        CommandWr(64 + 0x80)
        break;
    case 40 :
        CommandWr(20 + 0x80);
        break;
    case 60 :
        CommandWr(84 + 0x80);
        break;
    case 80 :
        CommandWr(0 + 0x80);
        Cursor[ActiveLcd] = 0;
        break;

    Cursor[ActiveLcd]++;
    LCD_OUT((c >> 4) + DEN +DRS);
    LCD_OUT((c >> 4) + DRS);
    LCD_OUT((c & 0xf) + DEN + DRS);
    LCD_OUT((c & 0xf) + DRS);
    return;

/* Write to LCD command register */
static void CommandWr(unsigned char c)

    LCD_OUT((c >> 4) + DEN);
    LCD_OUT(c >> 4);
    LCD_OUT((c & 0xf) + DEN);
    LCD_OUT(c & 0xf);
    return;

/* Initialize LCD panel */
void InitLcd(void)

    extern void Delay(char c);
    unsigned char b ;
    /* Put LCD in known state */
    Delay(20);
    LCD_OUT(3 + DEN);
    Delay(5);
    LCD_OUT(3);
    Delay(5);
    LCD_OUT(3 + DEN);
    Delay(5);
    LCD_OUT(3);
    Delay(5);
    LCD_OUT(3 + DEN);
    Delay(5);
    LCD_OUT(3);
    Delay(5);
    /* Set 4 bit mode */
    LCD_OUT(2 + DEN);
    LCD_OUT(2);
    Delay(5);
    /* 4 bit, 2 line, 4x7 matrix */
    CommandWr(0x2c);
    /* Display on, cursor off */
    CommandWr(0xc);
    /* Auto increment, shift right */
    CommandWr(0x6);
    /* Clear the LCD and set initial cursor address */

```

(continued)

Listing 2-continued

```

    ClearLcd();
    return;
}

/* General delay used by initialization function */
static void Delay(char c)
{
    while (c--);
    return;
}

/* Select Active LCD */
void SelectLCD(unsigned char c)
{
    if (c <= LCDs)
        ActiveLcd = c;
    return;
}

/* Assembler linkage: Output a byte over I2C bus */
static void LCD_OUT(unsigned char c)
{
    extern void Xmit_I2C_Byte(void);

    if (ActiveLcd == 0){
        B = c;
        ACC = Lcd0;
        Xmit_I2C_Byte();
    }
    else {
        B = c;
        ACC = Lcd1;
        Xmit_I2C_Byte();
    }
    return;
}

```

register is specified by holding RS low during the transfer sequence.

Cursor positioning consists of writing the cursor address into the appropriate global RAM variable and issuing a cursor positioning command to the LCD's command register. Of course, by now we know that what we see may not be what we get, so `PositionCursor` determines what adjustments must be applied to the cursor address before it can be transferred to the LCD. `ClearLcd` simply issues the clear command to the LCD's command register and zeros the appropriate cursor variable.

Finally, the `putc_har` function provides the low-level character output routine that is used by the stream I/O functions. This function is what all the other stuff exists for. All stream routines that output character data do so using the `putc_har` function. Newline characters are trapped at

this level, interpreted, and processed in a consistent manner that advances the cursor to the next physical display line. All other characters are simply transferred to the previously described `Write`. Note that no checking is performed for valid displayable data at any level of processing described here.

BATTERY CONTROL

If all you're using battery power for is occasional backup, then you can take some liberties in the design of the charging system. On the other hand, if the battery is the primary source of power to your system, it makes sense to devote a little more attention to getting the support circuitry right. This usually results in more power from a given battery and a longer operating life. In the case of the `ec.25`, the battery manager is centered around the bq2003 fast charge controller IC from Benchmarq Microelectronics.

Battery charging is accomplished by applying constant current with a fall back to a trickle level once a full charge is attained. Negative delta voltage is the method used to determine when fast charging is to be terminated with maximum time used as a failsafe backup. Although the bq2003 fully supports temperature-referenced charge termination, and this is what you'd want to use with a NiMH battery, this capability is not used in my configuration since I'm only supporting NiCds at the present time. Using voltage sensing to terminate fast charge eliminates the added components and interconnects necessary for temperature monitoring and simplifies the battery management circuitry and the battery pack itself.

Figure 2 shows the battery management card. All battery-related activities are controlled by U1, the bq2003 fast charge IC. The bq2003 is configured to control a linear constant-current source and is set up to use negative delta voltage as the charge termination criterium. These two selections are made by respectively pulling SNS to ground and DVEN to V_{cc} . TS and TCO are strapped in a way that inhibits temperature monitoring since it is not used in this configuration. The related TEMP output that is used to indicate temperature status via an LED goes unused also.

The charge initiation input, CCMD, is grounded, which enables automatic battery charging on either application of power or when a battery is connected. Tying TM1 and TM2 to ground selects a negative delta voltage holdoff time of 137 seconds and a fast charge safety time of 360 minutes. The holdoff time is the time after charge initiation that the battery is not monitored for maximum voltage and negative delta voltage. This prevents problems by allowing the battery's charging action to stabilize. The safety time is the maximum time that the battery will be held on fast charge before the charging cycle is terminated. This timeout should never be reached under normal operating conditions, which is precisely what it's all about: safety under abnormal operating conditions.

Attenuated battery voltage tapped from a divider made up of R14 and R10 is presented to the BAT input through an RC filter composed of R6 and C6. The divider is set up to develop a single cell voltage for the particular battery being charged. This voltage is constantly compared to the level at MCV that sets the maximum single cell voltage, in this case 1.8 V. If the BAT input becomes greater than the voltage on the MCV input, then charging activity is terminated. This is another safety feature.

Although the bq2003 can be used as a switch-mode constant-current source, here I am using it simply as a controller for an external linear current driver. The constant-current source uses a familiar circuit arrangement based on VR2, the popular LM317. The output current of 180 mA is set by R8 and, although not really a fast charge rate, it is adequate to ensure that a typical 600–850-mAH NiCd battery, composed of six AA cells, will attain a full state of charge in a reasonable period of time. This current source is controlled by U1 through switching transistors Q1 and Q2. Once U1 determines that the battery is on its negative voltage slope (negative delta voltage), current from VR2 is cut off. What remains is a small charge-sustaining trickle charge delivered through R4. This trickle resistor has another purpose as well. If the bq2003 determines that the attached battery is at a depleted state

of charge (less than 1 V per cell), fast charging is inhibited. In this case, it's up to the trickle resistor to bring the battery up a minimal voltage level before the bq2003 will energize the constant-current source.

If you look at the battery connector (J1) you will see there are a couple of positive pins: one outgoing and one incoming. When the battery is plugged in, these pins are shorted together since they go to a common connection at the battery's positive terminal. The reason for this is to prevent driving the unloaded output or VR2 (which rises to a relatively high voltage) directly into the main power feed of the system. Recall that last month I mentioned how the MAX639 micro-power switch-mode regulator doesn't do well with anything over 12 V on its input. Using two battery pins, one for input and one for output, is the simplest way of avoiding this type of problem. All you have to do is diode couple all the connections and everybody is happy. Finally, there is an extra unconnected pin on the battery connector for future use.

Having the ability to automatically do a discharge-before-charge on a NiCd battery can be particularly useful in preventing and correcting the voltage depression effect popularly known as "memory." An on-demand discharge cycle is started by pressing SW1, a momentary push-button switch that feeds into DCMD. When U1 detects a positive edge on DCMD,

the DIS push-pull output is turned on. This in turn drives the n-channel FET (Q3) into enhancement and effectively grounds R9, a 39-ohm power resistor. This load is kept connected until the battery is discharged to a nominal 1 V per cell. This level is indicated by the single cell voltage on the BAT pin falling below the reference level of $0.2V_{cc}$. When this voltage is reached, DIS is released and a normal charging sequence is started. All these different activities are indicated with just a single low-current LED (D3). Using sequences of pulses with varying duty cycles, a number of different events can easily be differentiated.

Next month: an I²C infestation. 📦

John Dybowski is an engineer involved in the design and manufacture of hardware and software for industrial data collection and communications equipment. He may be reached at john.dybowski@circellar.com.

SOURCE

For elements of this project, contact:

Mid-Tech Computing Devices
P.O. Box 218
Stafford, CT 060750218
(203) 684-2442

IRS

419 Very Useful
420 Moderately Useful
421 Not Useful

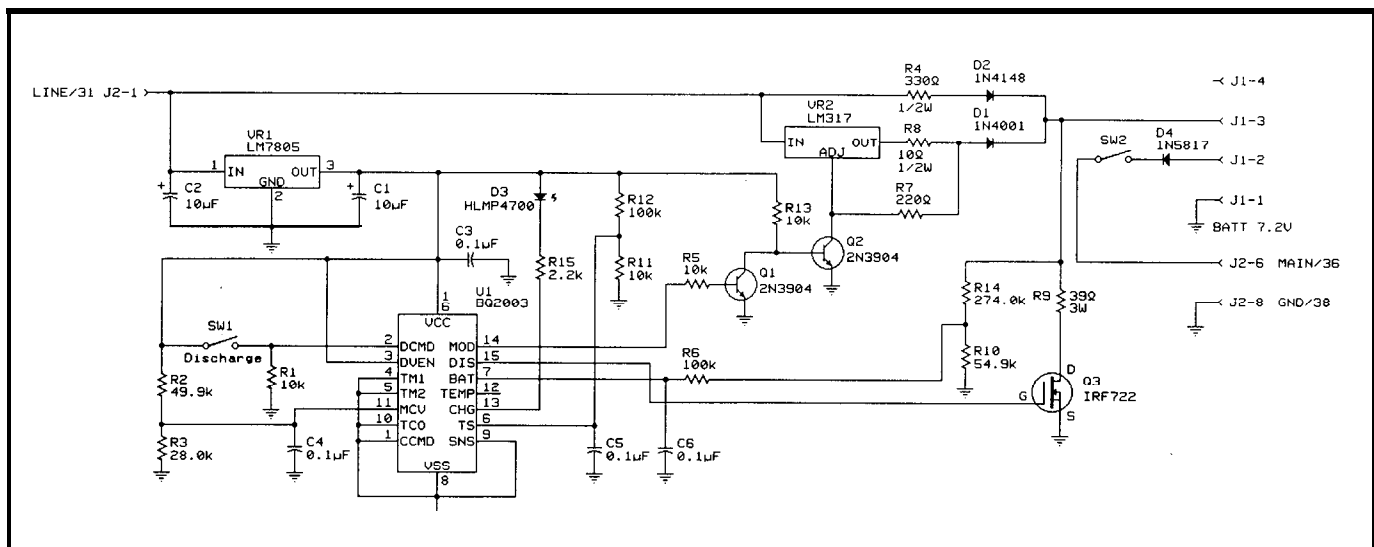
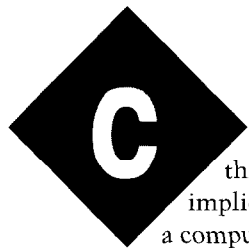


Figure 2—The battery management card specifically handles NiCds, but could handle other kinds of batteries with minor changes

PATENT TALK_{by Russ Reiss}



Computer security can mean different things to different people. To some it implies the process of restricting access to a computer system, its peripherals, programs, or data. To others the issues of "electronic signatures" and authentication/validation come to mind. Yet to others it means the use of computers (particularly microprocessors) to provide protection of other objects such as automobiles, homes, and valuable possessions. In this month's Patent Talk I'll look at seven patent abstracts that embrace all three of these areas.

Before getting into the mainstream, there's one patent I'd like to discuss that is somewhat different. The patent of Abstract 1 was retrieved as a result of one of the search processes I employed since it includes the words "microprocessor" and "security." It is an important patent for microprocessor-based system designers to be aware of because of its broad claims with regard to battery-backed RAM techniques. I have included the more verbose Claims

Text section which is available with the latest MicroPatent "U.S. Patent Search Claims & Abstracts" database. It appears that the novelty of this patent arises from the last "and" clause, namely the special startup security feature in which computer access is permitted only after an authorized user is identified. It is very unlikely that a patent could have been issued for any of the battery backup techniques or the security aspect alone. But by combining the two, a presumably unique device is created.

The next four abstracts describe patents that relate to means for restricting access to a computer. Abstract 2 by IBM (Great Britain) is perhaps the broadest. It covers the use of passwords stored in nonvolatile memory to determine the level of access permitted a given user. Crucial to the patent is the concept of multiple levels of accessibility in which a system manager has full access while users have restricted access. This restricted access is achieved in response to the user's password, which causes a unique boot configuration to occur, as was previously specified by the system manager. Applicability to both personal computers and networked systems seem possible here.

Carrying the password concept a step further, Dell Computers' patent in Abstract 3 stores a "code word identification" in a PAL. A program desiring restricted operation reads the code word from this PAL and compares



Patent Number 5204840
 Issue Date 1993 04 20
 Inventor(s) Mazur, Jeffrey G.
 State/Country CA

Title Means and methods for preserving microprocessor memory

Abstract Means and methods for preserving the RAM of an externally powered microprocessor on the occasion of a loss in external power. When the power loss is detected, a signal is generated which initiates a sequence to isolate the RAM and refresh it with an independent power supply. When main power is restored, the microprocessor is restored to its precise location at the moment of power loss. Shutdown, startup, and security routines are provided by software embodied in the system.

Ex Claim text A system for preserving at least the main random access memory (RAM) of a computer system having a power supply on the occasion of a sudden loss in power, said system comprising: a power loss detection circuit adapted to sense a loss in the power supply of said computer system and emit a signal causing a switchover circuit to isolate at least said main RAM from the computer system and cause at least said main RAM to receive its power from an independent power source; and a power switchover circuit responsive to said power loss detection signal to connect said independent power source to said memory; and an independent power supply comprising an external transformer power supply, a rechargeable battery, a voltage regulator, and a battery-charging circuit connected to said battery, said transformer power supply being connected to said voltage regulator and to said battery-charging circuit, said battery being connected to said voltage regulator for delivery of power to said switchover circuit; and a power-fall detection circuit comprising a precision voltage reference, means for measuring the input voltage to said computer system, and a comparator for initiating an NMI alert signal when said measured input voltage falls below said precision voltage reference; and software operatively associated therewith and responsive to said power loss detection circuit signal to initiate a shutdown routine when said signal is activated and to initiate a restart routine when said signal is deactivated; and security means for interrupting a normal startup sequence of said computer system to invoke a security procedure, whereby further use of said computer system would be allowed only after identifying an authorized user.

U.S. Refs 4578774 4631701 4718038 4788661 4815032 4823308 4897631 4901283 4959774 4977537 5018096

PATENT TALK


it to one stored in the program. Access is permitted only if the two code words match. This scheme is far reaching. The PAL could be made an integral part of the system by soldering it directly to the motherboard. Special versions of operating systems, application code, device drivers, and the like can then be restricted to running only on a system so equipped with such a PAL. In contrast to a password which restricts all but one (or a few) user(s), this code word restricts the execution of software run by *all* users.


Appropriately named, Computer Security Corporation's patent in Abstract 4 is a plug-in expansion board for a PC. It contains a number of components including a BIOS extension EPROM; scratchpad RAM; EEPROM for storing passwords, audit trail log, encryption flags and the like; a data encryption/decryption chip; and a clock chip. Along with special software, these devices become security tools that can restrict access to the system itself only to users with valid passwords, keep data written to disk secure through encryption, prevent unauthorized disk formatting, prevent file copying, and even prevent circumvention of the security system itself during boot. Although not mentioned,

the system could also restrict access to certain peripherals (e.g., CD-ROMs or modems) only to authorized users.

Finally in this category, Abstract 5 covers a patent that has the specific and singular function of restricting user access to various peripheral devices on the system. The typical password is requested during boot, which retrieves user authorization information from memory. Thereafter, every attempt to access a peripheral over the I/O channel is monitored by hardware and trapped if not permitted. Trapping involves setting a flag indicating security violation as well as blocking the access over the system bus. Such a device would require access to the system bus, and hence could take the form of an expansion card. However, the inventors chose to refer to it as a "module," which would imply that they also see it taking a form which could be mounted directly to the motherboard.

Shifting our attention now to another area, the next abstract relates to the topic of "electronic signatures." It has long been promised that, in this age of a "paperless society," we would be able to replace the handwritten signature with an electronic equivalent: a means for

Patent Number	5265163	
Issue Date	1993 11 23	
Assignee	International Business Machines Corp.	
Inventor(s)	Golding, Victor G., Speirs, Gregor H.	
State/Country	GBX	
Title	Computer system security device	
Abstract	A computer system having a power-on password stored in nonvolatile memory wherein entry of the power-on password by a system manager permits access to all of the computer functions. The system also has the facility of at least one additional password held in nonvolatile memory, wherein entry of the additional password by a user permits the system to boot in a manner preselected by the system manager. Preferably there are available a plurality of additional passwords providing at least two different levels of security access to the system.	
U.S. Refs	4757533 5012514 5058164 5115508 5204966	

Patent Number	5,099,516	
Issue Date	1992 03 24	
Inventor(s)	Durkin, Michael D.; Stewart, Greg N.	
State/Country	TX	
Assignee	Dell Corporate Services Corporation	
Title	Digital computer code word identification system	
Abstract	A digital computer system has a central processor unit (CPU). A computer program, entered into the digital computer system for execution thereof, has a program code word embedded at an arbitrary location therein. An addressable programmable array of logic (PAL) is operatively connected to the CPU for receiving a READ signal originated by the CPU at the address of the PAL, the PAL being programmed to output a portion of a preset array code word in a response to the READ signal, and to output the remainder of the array code word in segments in response to subsequent READ signals at the same address. A data bus, connected to receive and transmit the portion and remainders of the array code word to the CPU for comparison with the program code word. The program code word and the array code word are compared and, if identical, permit use of the program and do not permit use when the program code word and the array code word are not identical.	

PATENT TALK



Patent Number 4,757,533
Issue Date 198807 12

Inventor(s) Allen, Michael J.; Langlois, John
State/Country IL
Assignee Computer Security Corporation

US References 2 7 , 2 5 1 3,368,207 3,566,362 3,573,855 3,771,146 3,828,327 3,931,504 4,040,034 4,078,152 4,095,738
4,095,739 4,121,523 4,177,510 4,246,638 4,255,786 4,262,329 4,19,079 4,386,233 4,395,628 4,464,566
4,519,032 4,558,176 4,609,996 4,644,493 4,652,990 4,654,821 4,658,093 4,672,182

US Class 380125 380/23
Int. Class H04L 9/00

Title Security system for microcomputers

Abstract A security system for a personal computer in which hardware and software are combined to provide a tamper-proof manner of protecting user access and file access. The hardware component of the system is an expansion board for insertion into an expansion slot of the PC and has a first EPROM chip containing four portions of machine code for initializing system function calls and for establishing the proper boot-processing of the PC; a second RAM chip serving as scratchpad memory; a third EEPROM chip storing passwords, audit trail log, protection and encryption system flags, and user-access rights; a fourth automatic encryption and decryption chip for files of the PC; and a fifth clock chip for the audit trail. The software component includes a batch file that runs a program in conjunction with the machine code on the EPROM of the expansion board ensuring access is gained only for valid users. The code on the EPROM monitors all DOS 21 H file handling function calls, and initializes the 7CH interrupt vector for allowing the security system to access DOS and files thereof. During boot processing, the 10H video interrupt handler is monitored to prevent circumventing the security system. Hard disk format protection is also provided by monitoring of the 13H interrupt function calls. Files may also be created that may not be copied.

verifying that a document was executed by only one specific individual. This capability is of particular importance and convenience if the authentication must take place over a distance via telecommunication. Although somewhat scant on detail, this patent could serve as a starting point for someone wishing to pursue developments in this field.

The final abstract relates to security associated with the use of IC Cards (also called "identification cards," "chip cards," "memory cards," and such]. Abstract 7 discusses a means of limiting the authorization period of such a card by "scrambling identification card information with time information." The system which reads the card can recover the time information and deny access if the card has



Patent Number 5202997
Issue Date 199304 13
Assignee Isolation Systems Limited
Inventor(s) Arato, G. Peter
State/Country CAX

Title Device for controlling access to computer peripherals

Abstract An access control module restricts access to a computer system to authorized users and selectively controls each user's access to associated computer peripherals such as data storage units, printers, and communications equipment. During startup of the computer system, a microprocessor associated with the module invokes a software routine that requests entry of a valid user identification code. In response, the microprocessor retrieves from a main nonvolatile storage unit, prerecorded information regarding the user's authority to access each of the peripherals and loads the information into a secondary storage unit comprising random access memory and an address decoder adapted to retrieve data therefrom. Thereafter, the address decoder responds to each peripheral address signal generated in the input/output channel associated with the computer system and retrieves from the secondary storage unit the access information relating to the peripheral identified by the address signal. A latching circuit generates and maintains a signal indicating a violation of computer security if the retrieved information indicates that the current user of the computer system is not authorized to access the selected peripheral. The microprocessor responds to the signal indicating security violation by applying signals to the input/output channel which interrupt the operation of the computer system and interfere with access to the selected peripheral.

U.S. Refs 3368207 3398405 3473159 3508205 3585606 3725872 3872444 4000487 4264782 4601011

PATENT TALK

Patent Number	5,163,091
Issue Date	1992 11 10
Inventor(s) State/Country	Graziano, James M.; Dziewit, Halina S. c o
US References	4,453,074 4,568,936 4,633,036 4,807,287 4,893,338 4,893,340
US Class Int. Class	380/25 380/49 3401825.34 H04L 9132
Title	Knowledge-based system for document authentication (apparatus)
Abstract	The document authentication apparatus provides document authentication and authenticity capability. Document authentication requires the person to be charged, apply an authentication mark on the document indicating intent to authenticate the document. This requirement is analogous to a signature on a printed document and is implemented in the document authentication apparatus electronically through the use of both hardware and software. A program immediately checks the identicalness of the document at the transmitting and receiving station through a high-speed comparison, locks in the document such that no modification can occur, and awaits authentication handshakes from the two endpoints. Such authentication is real-time and can be both hardware and software executable (i.e., password and physical confirmation).



Patent Number	5,067,155
Issue Date	1991 11 19
Inventor(s) State/Country Assignee	Bianco, James S.; Madsen, James T.; Ceppetelli, Michael; Fahy, John S. CT Control Module Inc.
US References	4,599,489 4,694,492 4,799,258 4,809,326 4,885,788 4,944,008 4,951,249
US Class Int. Class	380/25 380123 380/3 G1 1 B 23128 H04K 1 /00
Title	Method and means to limit access to computer systems
Abstract	In a preferred embodiment, a method of limiting access to computer systems which method includes scrambling identification card information with time information so that the resulting code can be used for only a limited period of time, thus preventing unauthorized persons from using the code at a later time. In another aspect of the invention, one or more p-metal shields are embedded in an identification card, thus identifying the card as being valid and also providing means to indicate when a valid card is being removed from a card reader.



expired. Presumably, "scramble" implies that the time information is not readily observable by someone wishing to circumvent the protection by altering the card. □

Russ Reiss holds a Ph.D. in EE/CS and has been active in electronics for over 2.5 years as industry consultant, designer, college professor, entrepreneur, and company president. Using microprocessors since their inception, he has incorporated them into scores of custom devices and products. He may be reached at russ.reiss@circellar.com or 70054.1663@compuserve.corn.

SOURCE

Patent abstracts appearing in this column are from the Automated Patent Searching (APS) database from:

MicroPatent
25 Science Park
New Haven, CT 06511
(203) 786-5500 or (800) 648-6787

MicroPatent databases include the abstract-only APS version; FullText, which contains the entire patent without drawings; PatentImages, for the complete patent listing including drawings; and other specialized databases for just chemical, computer, or European patents.

422 Very Useful 423 Moderately Useful 424 Not Useful

CONNECT TIME

conducted by Ken Davidson

The Circuit Cellar BBS
300/1200/2400/9600/14.4k bps
24 hours/7 days a week
(203) 871-1988—Four incoming lines
Internet Email: @circellar.com

With all the talk of Internet I've been doing /ate/y, I'm often asked how someone can go a step further than the Email we offer and gain full net access. Most people get full access through the university they work for or attend, or through their employer who has full network connectivity coming into the office. I normally have to tell people without these options to look into one of the commercial systems that offer full access such as BIX or Delphi. However, another alternative has recently come to my attention that will cost you the same as it does for just the Email we provide: nothing but the cost of a long-distance phone call.

The International Internet Association is offering free accounts to anyone who asks. The only drawback is you must make a call to either Washington D. C. or New Jersey to get to their access point. They also offer an 800 number at very reasonable rates. They are working on making more access points available across the country, so you may eventually end up with a local node, so access would cost nothing at all. Features include 14.4 kbps, telnet, ffp, Archie, Gopher, WWW, and more.

There are no hidden hooks or gofchas that I can see, so it's certainly worthwhile to check out. For more information, contact IIA, 2020 Pennsylvania Ave. N. W., Ste. 852, Washington, DC 20006, (202) 387-5446, or Email to info@iia.org.

With all the melting snow and spring rains associated with April, people are bound to end up with wafer in the basement. Detecting that water early can help prevent major damage, so I decided to devote most of the column this month to a rather amusing and interesting discussion we recently had on water defectors. The second thread deals with the ratings found printed on the side of most relays.

Water Detector

Msg#:19995

From: RICK RETTER To: ALL USERS

I have a need for some sort of simple safe circuit that can detect water. Can anyone out there help me? Thanks for any suggestions.

Msg#:20006

From: PAUL PETERSEN To: RICK RETTER

How much water? Gallons? Misty droplets? It makes a difference in how we proceed.

Msg#:20199

From: JEFF BACHIOCHI To: RICK RETTER

I've seen some cheap water sensors made from a piece of sponge which, when wet, expands to press a tiny switch. As the sponge dries out it shrinks back, releasing the switch. Real cheap. Varying the size will determine the amount of water necessary to press and release the switch.

Msg#:20316

From: RICK RETTER To: JEFF BACHIOCHI

I guess I wasn't specific enough. I need a detector for a water sump pit. When the water is high, I turn on a pump via the PC parallel port, and when the water level decreases, I turn the pump off. I can't use the normal float switch that come with sump pumps since the "travel" is too much.

Msg#:20353

From: JEFF BACHIOCHI To: RICK RETTER

Please explain further, normally when the pump goes on you get a large reduction in water level and when the pump turns off the back wash raises the level. This can cause oscillations in the pump if the hysteresis isn't large.

Msg#:20363

From: ED NISLEY To: RICK RETTER

Use two switches: a high limit and a low limit. Attach a flag to the float and have it activate each switch in turn. Run two bits into the PC and proceed as before.

Actually, if you've got a PC in the loop, I'd add a little self-checking to the switch. Use a pair of optical or metal-proximity switches and make the flag long enough so it can activate both of the switches when the water level is between the limits. Then the PC can verify that the state transitions make sense:

low = both were on: high now off and low now on

filling = low was on alone: both now on

high = both were on: high now on and low now off

draining = high was on alone: both now on

Note that you need to keep track of the previous switch state. It'd be a good idea to add some timeouts to this so if the pump goes on and the water level doesn't drop you can initiate manual bailing..

CONNECT TIME

But, seeing as how you've got a PC in the loop, I'd also add a mechanical emergency backup switch about half an inch above the upper limit: it directly powers the pump without any of this computer nonsense. If the PC goes out to lunch, you can at least ensure that the basement won't fill up...

Msg#:20475

From: JIM NELSON To: RICK RETTER

Aww, these guys wanna make it too simple! You've already tied up a PC, so why don't we go all the way!

Oops-nonrhetorical question #1: What happens when the PC is turned off?

But since you want to get complicated, *Sensors* magazine is packed with high-tech fluid level sensors (laser, ultrasonic, microwave, etc.) that will nicely complicate this pump control. You'll also want to consider your definition of water level, since the transition between water and air is kinda vaporous.

OK, to get a line on that, your new and improved detector will operate as a function of vapor pressure and humidity, so you'll need both humidity and temperature sensors, and if the water isn't absolutely pure, we'll have to know what it is, so add an ion mobility spectrometer (what?) so you can pinpoint that vapor pressure. Factor in air density, a function of altitude, local acceleration of gravity, the number of holes in your dryer's lint trap, and the proximity of the moon, Alpha Centauri, and the Mars Observer.

Clearly this is a job for fuzzy logic.

Now with fuzzy running on the Baby Cray humming in the other corner of your basement, you'll happily compensate for some of the smaller effects on your pump control. How many people can factor in the very small, yet provably finite probability of a lightning strike which would completely obviate the need for even bothering with the pump? A job for fuzzy!

We can't forget the smoke and fire detectors you'll need to protect yourself against this seething mass of solution, nor automatically triggered Halon fire extinguishers. And active noise cancellation to quiet this buzzing mess.

Personally, I would rather go swimming in a lake than in a basement, especially one crammed with molten electronic junk glowing and blowing smoke bubbles.

P.S. Don't forget a moisture detector-so you'll know when it all fails.

<sly grin>

Msg#:20869

From: JOHN CONDE To: JIM NELSON

Well if you want to go slightly less than all the way, I've seen three basic (simple, cheap) water-level detectors.

The first is a float switch, which you don't like.

The second is an all electric probe where you have two contacts close together at the level at which you want to trigger supplied with a voltage from a supply will almost no amperage. When the water level reaches the probe, voltage will go to zero. That should be easy to detect. (Note this method doesn't work with pure water [an insulator], but basement water should work fine.) To construct it, put a 1-meg resistor between the power supply and the plates, then measure voltage from the plate side of the resistor. The higher the voltage, the further apart the plates can be (experiment]. Also, you will have to put some sort of strainer around the probe to prevent solids from floating in and shorting the plates, plus you have to clean the regularly.

The third and lowest maintenance version takes a little more construction. Inside a sealed tube you put a temperature sensor and a small heater, then mount the probe at the height you want to sense water. Since water removes heat much faster than air, the equilibrium temperature for the probe (and the resistance through the sensor) will change drastically. Also the time it takes to heat or cool the probe will give you some built-in hysteresis. Of course, these descriptions are short of specifics, but I hope they are helpful.

Msg#:21357

From: CHRIS TYLKO To: JOHN CONDE

There's a fourth simple way. My submersible sump pump has a pressure sensor. It's just a diaphragm that activates a microswitch. A very small tube runs through the AC power cable whose plug end must be open to atmospheric pressure. As the water level goes four or so inches above the top of the pump, it activates, emptying the pit until the pump sucks air; by then the diaphragm lets go and turns the pump off. In ten years, I've only had to replace the microswitch.

Ironically, I guess, the pit is right below my HCS control center (which is in the wine cellar), but there are—and never will be—any connections between the system and the pump.

Msg#:21469

From: JOHN CONDE To: CHRIS TYLKO

My list was not really intended to be exhaustive, I'm quite sure there are even more simple ways (and probably some astoundingly complex ones). The pressure switch design is a good one. I would guess that most of the automatic pumps use one. I agree about not controlling the pump with a computer; some things are just too important to trust to a computer.

CONNECTIME

Msg#:21610

From: JEFF BACHIOCHI To: CHRIS TYLKO

Control the pump through the HCS! Never, is right! However you may wish to keep a record of pump activity or an emergency high-level sensor for those times [if/when] the pump fails!

Msg#:24062

From: CHRIS TYLKO To: JEFF BACHIOCHI

Actually a very good idea, considering when the pump did fail it cost me a few sheets of 4x8 presswood.. .and you know how quickly that stuff soaks up water!

What, in your opinion would be the simplest and cheapest detector to hook up to the HCS!

Msg#:24189

From: JEFF BACHIOCHI To: CHRIS TYLKO

I like the tube with a closed bottom idea containing multiple (at least two) hall effect switches. A Styrofoam doughnut [containing a magnet] is placed on to the tube. The float rises and falls with the water level, the sensors detect it passing various points, the electronics stay dry, and you decide how to interpret the data (turn on alarm, pump, or call flood control central!)

Another cheap detector would consist of two metal thumbtacks (the contacts) pushed into the jaws of a spring wooden clothespin (the force). The contacts (normally open) held apart by an aspirin (the detection device). Water would melt the aspirin and the contacts would close.

Assuming you have a pump, I would use the float switch on the pump unit as an input.

Assuming you didn't, I would get a piece of Styrofoam and mount a microswitch on the top surface (the kind with a cantilever arm). Place the arm through a hole in the center of the foam piece and adjust the switch's position so that if the Styrofoam raises off the floor at all, the switch closes.

Msg#:35154

From: GARY JOHNSEY To: CHRIS TYLKO

I once needed to maintain the water level in a bucket, quick and cheap. So I pulled the water level sensor and water control solenoid out of an old clothes washer. The switch contacts were sealed and not directly exposed to moisture; probably one reason they outlasted the ten-year-old washer.

However, I like Jeff's Styrofoam float idea, especially if you can seal the contacts. Maybe a glass-encapsulated reed switch? Put the magnet on the float and either paint it or dip it in that liquid rubber like stuff used to insulate pliers handles. Or, maybe not. Just a thought.

Msg#:35345

From: TOM NICKEL To: GARY JOHNSEY

I was just looking a National's 1993 Linear Application Specific ICs data book. They have three components that can measure/detect fluid levels: LM903, LM1042, and LM1830.

I can suggest one method to *not* use: Some time back, a few of my friends were living in a studio which had a notoriously leaking/flooding basement problem. We set up an old snare-drum stand with a piece of coat hanger wire, chunk of Styrofoam, microswitch, and a "yelping" automotive-type intrusion alarm, borrowed from their truck. The idea, of course, was that rising water would case the "float" to float and trigger the switch. Problem was, the automotive alarm device had poor immunity to power line spikes (I guess) because it would false trigger, usually in the wee hours of the morning. Needless to say, the system was quickly disarmed and the basement promptly flooded! (My engineering skills have improved a tad since then, I don't use as many coat hangers in my new designs.)

Msg#:37257

From: DAN HOPPING To: CHRIS TYLKO

A simple moisture detector I have used in the past is to press two wires together using one of those magic expanding sponges (you know the ones that look like a sheet of paper, then when you put them in water they pop into a full sized sponge). They expand with a remarkable (relatively) amount of force and have no problem closing the contacts. They have an almost instant response to water. I made a few fixtures up with a battery and some surplus Sonalerts. Cheap and simple. Great for under the sink and behind the fridge. I *never* trust my own plumbing connections...

By the way, you can reflatten those little sponge actuators by putting them in a vise and letting them dry out again.

Msg#:22556

From: PELLERVO KASKINEN To: JOHN CONDE

I happen to have an experience with the water-level sensing using a couple of electrodes and a resistance-sensing trip circuit. Turned out to be exactly what you mention about the frequent need for cleaning! There were a couple of reasons in my case. One was that the water was hot and tended to condense on the insulators of the electrodes. The other one is that the DC principle causes electrolytic action on the electrodes. That action tends to anodize one of them and make it something similar to a low-leakage capacitor.

The commercial water resistance measurement electrodes are made of platinum and they are *always* operated on AC, not DC, which makes the tripping circuit a little more complicated. Not necessarily much-think of

CONNECTIME

the indicators on commercial water purifiers (deionizer cartridges). You just have a 115-VAC supply in series with a current-limiting resistor, a neon bulb, and the sensing electrodes. If the water becomes conductive (or the level reaches the electrodes), then the light comes on. You could glue a photocell to the neon bulb and have the sensor.

A better, but more complicated, principle is to use the high dielectric constant of water and, instead of resistance, measure the capacitance of the probe. While typical insulating materials have a dielectric constant of about 2-5, water exhibits a high constant of 71 and is therefore quite easy to detect by the capacitive principle.

Another recently popular method is to use an acrylic (Perspex, Lucite, etc., trade names) rod with one prismatic or conical end and one flat end. You shoot a beam of light into the flat end. If it comes back, the conical end is not immersed. If the light is lost, the conical end is immersed in something more optically dense than air, (i.e., in water). This same principle is used on some fuel gauges on kerosene lamps and/or heaters, with a staggered prismatic end on the rod. You see a different pattern depending on which ends are immersed and which are up in the dry.

Msg#:23415

From: JIM NELSON To: PELLERVO KASKINEN

I do like the index-of-refraction stunt that's behind the immersible acrylic plastic sensors you described. Sounds like something that could be nicely packaged-an LED and a detector/amp die cast in the same plastic package-if it could be sealed, and if water-borne contaminants and organic life (crud) accreting to the immersed surfaces didn't become a problem. Are such units available off the shelf?

Msg#:23739

From: PELLERVO KASKINEN To: JIM NELSON

As to the fouling, if you invent something that those little critters in water would not foul, you would be a rich man. But there are differences and then there are differences! Acrylic by its water absorption characteristics is not the greatest. Polystyrene might actually sometimes beat it. But of course, glass is superior and quartz is the best as far as I know. However, I have not kept an eye on the commercial availability of these devices. I would think there must be some sources, as this is not such a novel idea any more. My first exposure to them was already more than 20 years

BCC52 BASIC -52 COMPUTER/CONTROLLER

The BCC52 controller continues to be Micromint's best selling single-board computer. Its cost-effective architecture needs only a power supply and terminal to become a complete development system or single-board solution in an end-use system. The BCC52 is programmable in BASIC-52, (a fast, full floating point interpreted BASIC), or assembly language.

The BCC52 contains five RAM/ROM sockets, an "intelligent" 27641128 EPROM programmer, three 8-bit parallel ports, an auto-baud rate detect serial console port, a serial printer port, and much more.

PROCESSOR

- 80C82 8-bit CMOS processor w/BASIC-52
- Three 16-bit counter/timers
- Six interrupts
- Much more!

MEMORY

- 48K RAM/ROM, expandable
- Five on-board memory sockets
- Either 8K or 16K EPROM

Input/Output

- Console RS232 - autobaud detect
- Line printer RS-232
- Three 8-bit parallel ports
- EXPANDABLE!
- Compatible with 12 BCC expansion boards

To Order Call 1-800-635-3355

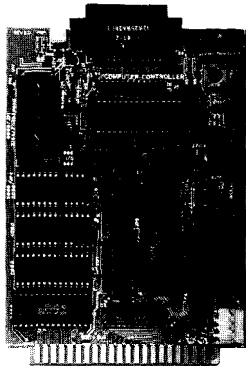
Tel: (203) 871-6170

Fax: (203) 872-2204

BCC52	Controller board with BASIC-52 and 8K RAM	\$189.00	Single Qty
BCC52C	Low-power CMOS version of the BCC52	\$199.00	
BCC52I	-40°C to +85°C industrial temperature version	\$294.00	
BCC52CX	Low-power CMOS, expanded BCC52 w/32K RAM	\$259.00	

CALL FOR OEM PRICING

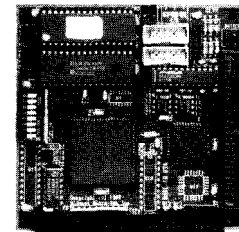
MICROMINT, INC. 4 Park Street, Vernon, CT 06066
in Europe: (44) 0285-658122 • in Canada: (514) 336-9426 • in Australia: (02) 888-6401
Distributor Inquiries Welcome!



We're Small, We're Powerful, And We're Cheaper.

MMT-188 EB

- 2 serial I/O ports
- 3 programmable parallel I/O ports
- 1 Meg RAM/ROM capable
- powerfail detect interrupt and reset
- counter-timers
- watch dog timer
- expansion connector



Only \$191⁰⁰ (Qty 100)

ALSO AVAILABLE: MMT-Z180, MMT-196, MMT-HC11, MMT-EXP

In fact, you'll get the best product for about half the price. If you're interested in getting the most out of your project, put the most into it. For the least amount of money.

Call us today for complete data sheets, CPU options, prices and availability.

Custom Work

Welcome. Call or fax for complete data sheets

2308 East Sixth Street
Brookings, SD 57006

Phone (605) 697-8521

Fax (605) 697-8109



WE'RE SMALL BUT WE'RE POWERFUL

CONNECT TIME

ago, although it was only for visual feedback rather than for electronic signaling. It was, by the way, used in a kerosene heater by the British subsidiary of the Aladdin company, which is based in my former home town, Nashville. I think the small critters that live in water do not like the kerosene too well..

Msg#:24570

From: LARRY G NELSON SR To: PELLERVO KASKINEN

I used the index-of-refraction technique in 1975 to measure height in a column of fluid. Picture a long line of triangles in a tube with the flat side along the wall of a long tube. The LEDs were on the top half of each triangle with the detector on the bottom half. This was cleaned up and fed to some TTL glue to encode the binary data and used to indicate the level to a bar graph as well as to the computer running the equipment. It worked quite well and was considerably cheaper than the competition at the time while looking high tech.

Relay ratings

Msg#:49468

From: STEVEN SUN To: ALL USERS

I have a question about relay contact ratings. I am trying to control a signal line in a car. According to the car manufacturer, the maximum amperage on that signal line is 10 amps. I am wondering if I can use a relay with a contact rating of 1 amp at 125 VAC to control this signal line. My understanding of the contact rating is that it will handle a maximum of 125 watts. Since the car signal line is limited to 12 watts, that relay should be able to handle it. By the way, the relay contact is normally closed, so it should be able to handle the 10-amp current flow almost continuously.

Can anyone please explain to me what exactly does the contact rating on a relay mean? Any assistance will be greatly appreciated.

Msg#:49517

From: ED NISLEY To: STEVEN SUN

Contact ratings can be tricky, but the rule of thumb is that you may not exceed any part of the rating.. no matter how innocent it may seem.

For example, if the rating is given in terms of AC volts you cannot apply any part of that spec to DC voltages or currents. The reason is simple: the arc created across contacts as they open will self-extinguish if it's driven by AC current, but it'll just keep burning away on DC. Small relays have roughly the same AC and DC current ratings at

a given voltage, but the rating is voltage-dependent: higher voltages mean lower currents. The ratings for larger relays can differ by orders of magnitude, so don't extrapolate upward!

If a relay contact is rated at 1 amp at 125 VAC, that means if you apply 125 volts AC, the contact may not carry more than 1 amp. It does *not* mean the relay can handle any combination of voltage and current that equals $(1 \text{ A}) * (125 \text{ VAC}) = 125 \text{ volt-amps AC}$. That prevents you from trying to run 125 A at 1 VAC.. which is obviously improper.

There is a difference between the "carrying" rating and the "interrupting" rating, but for typical small relays you only see the latter because you're using the relay to open or close a power circuit. If you can *guarantee* that the relay is always cold-switched (meaning the contacts open or close *only* when they're not carrying any current) then you can use the "carrying" rating, but in most applications the "interrupting" rating is the one you've got to use.

Bottom line: if you want to switch a 12-V circuit carrying 10 A, you need a power relay rated for service at 12 V and 10 A.. anything else just won't work.

I bet that's what you figured, but you were hoping there was a loophole.. right?

We invite you call the Circuit Cellar BBS and exchange messages and files with other Circuit Cellar readers. It is available 24 hours a day and may be reached at (203) 871-1988. Set your modem for 8 data bits, 1 stop bit, no parity, and 300, 1200, 2400, 9600, or 14.4k bps. For information on obtaining article software through the Internet, send Email to info@circellar.com.

ARTICLE SOFTWARE

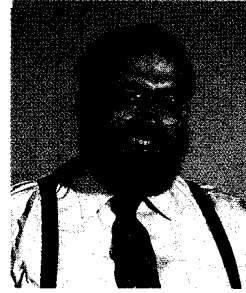
Software for the articles in this and past issues of *The Computer Applications Journal* may be downloaded from the Circuit Cellar BBS free of charge. For those unable to download files, the software is also available on one 360K IBM PC-format disk for only \$12.

To order Software on Disk, send check or money order to: The Computer Applications Journal, Software On Disk, P.O. Box 772, Vernon, CT 06066, or use your VISA or Mastercard and call (203) 875-2199. Be sure to specify the issue number of each disk you order. Please add \$3 for shipping outside the U.S.

425 Very Useful 426 Moderately Useful 427 Not Useful

STEVE'S OWN INK

In the Citadel



O fiber optic metropolitan area networks promise to bring me more information bandwidth than was previously ever conceived possible (or desired). But do I really want all of that bandwidth and the "intelligence" that is imposed on the otherwise innocent carriers?

Remember when you had to climb up on the roof to turn the antenna just so to get the TV to receive the four or five broadcast channels that were available in your area? It was a bit surreal having a receiver capable of picking up more than 60 channels where only three or four would have anything on them. Even though I knew there were only four channels, periodically I would methodically tune to each one hoping to catch a new station, but finding none, would always return to my favorite channel anyway. I guess back then, one channel would have been enough for me.

Next came the proliferation of cable systems and the rise of the superstations. The providers called this progress, because now I had more than 100 channels, and most of them carried some kind of "intelligence." Intelligence in this case being defined as an analog waveform modulated on the carrier wave. I don't want to quibble over terms, but the trouble is most of the "intelligence" on this plethora of offerings is just not "intelligent." I suppose I must be on an extreme edge of the demographics curve, but I just don't appreciate QVC, E!, DAETC, and all the other mindless, narrow-focus channels that are being offered. Even though I have all these channels to choose from, I still am mostly confined to a small number of channels that I consistently view.

Now the providers want to offer me hundreds(!) of channels, with interactive TV to boot. I have to ask myself if I really want, or need, this kind of service. It's a sad statement indeed when someone who is as much a technophile as me ponders whether or not to take part in emerging technology. I guess I'm not convinced I need a few hundred channels more of the trash that the providers consider entertaining. It is interesting that the ads for these systems promise interactive learning, video conferences, and an all-electronic democracy, when I have an inkling it will more likely contain shows like The Interactive Dating Game, Holographic Lifestyles of the Obscenely Rich, not to mention too many channels of interactive psychics, video sex, and the Nintendo channel.

Bah! Who needs it?!?

I suppose the only thing this new wave of bandwidth happiness could offer me is the chance for more intellectually stimulating programming, but I don't think this is likely to happen as long as the same cloistered crew of media pundits who brought you too much of the Bobbit trial rethink what their mission is. I mean, do they want to educate us or stupefy us? Some great thinker once said that an educated person is more likely to become free; it scares me to think that educating us is the last thing the moguls and the kingpins of the media machine want to do....

A handwritten signature in cursive script that reads "Steve".