

CIRCUIT CELLAR **I N K**<sup>®</sup>

# THE COMPUTER APPLICATIONS JOURNAL

May 1993 — Issue #34

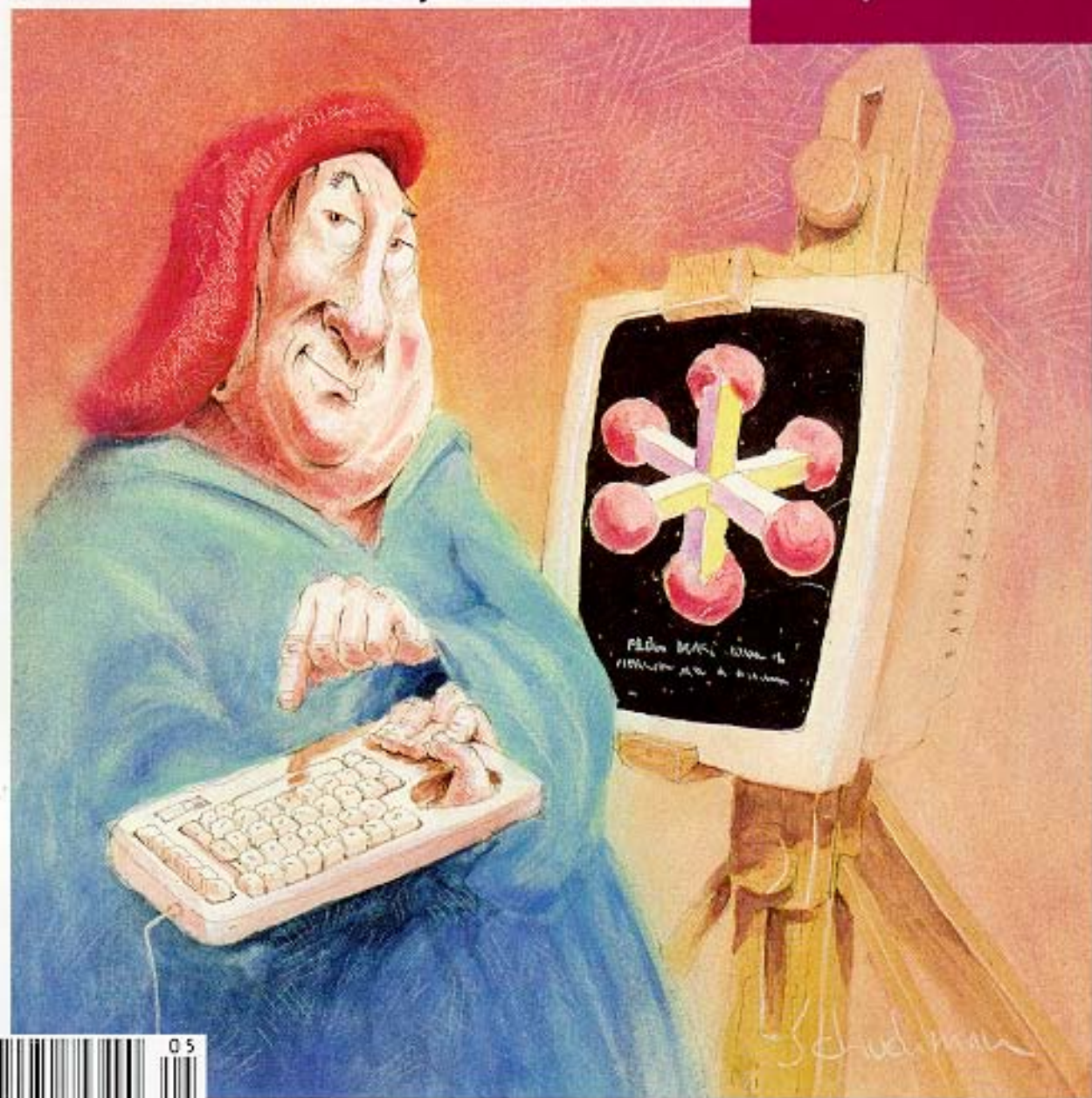
## GRAPHICS & VIDEO

Closed Caption Decoding  
with a 68HC05

Sony's New Mini Disc

Custom Graphics LCD  
Controller Chip Design

Vector Math for  
3D Graphics



\$3.95 U.S.  
\$4.95 Canada



## A Match Made in Heaven

HOW perfect a marriage of graphics and video: closed caption decoding and display. Pioneered by the public broadcasting system, closed captioning has been growing in popularity in recent years. In fact, this system that aids people with hearing impairments enjoy television with the rest of us has become so prolific that Congress has mandated that, as of July 1, 1993, all television sets sold in the US. with screen sizes 13" and bigger must include closed caption decoding circuitry.

After having a chance to experiment with the Vertical Blanking Interval Explorer project that won second place in our last Circuit Cellar Design Contest, I have seen first hand just how much of today's television programming already includes closed caption information. If you buy a new TV in the next few months, you, too, will be able to see how useful a system it really is.

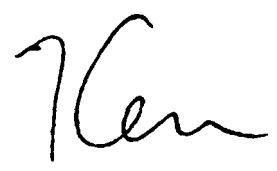
Motorola has taken a giant step in helping manufacturers include closed caption decoding in future televisions at minimal additional cost. Almost all of today's larger televisions already include a microcontroller and on-screen character display. The new 68HC05CC1 microcontroller also adds caption decoding and, with minimal additional components, directly drives the red, green, and blue CRT guns with superimposed captions. Be sure to check out our first feature article that describes the chip's design and use.

Next, when rendering three-dimensional graphics on a two-dimensional display device, a fundamental understanding of vector and matrix mathematics is crucial. Find out what's involved with such math and how to do it.

In our final feature article, David Erickson combines custom chip design with an off-the-shelf LCD display to off-load much of the graphics processing from the microcontroller to the display controller. You, too, can make a chip.

In this month's embedded '386SX installment, Ed adds some timers to the system as he continues to lay the foundation for his system. Jeff does some hands-on testing of the myriad kinds of batteries found on the market today and makes some suggestions for which kinds are best used in what applications.

Tom goes over the details of Sony's exciting new Mini Disc. Not only does this system promise to replace traditional cassettes, but it should be a boon for computer mass storage and archival. John takes a look at Signetics' I<sup>2</sup>C bus and its usefulness in embedded systems. Finally, Russ covers patents dealing with this issue's graphics and video theme.



# THE COMPUTER APPLICATIONS JOURNAL

**FOUNDER/EDITORIAL DIRECTOR**  
Steve Ciarcia

**EDITOR-IN-CHIEF**  
Ken Davidson

**TECHNICAL EDITOR**  
Michael Swartzendruber

**ASSOCIATE EDITOR**  
Robert Rojas

**ENGINEERING STAFF**  
Jeff Bachiochi & Ed Nisley

**WEST COAST EDITOR**  
Tom Cantrell

**CONTRIBUTING EDITORS**  
John Dybowski & Russ Reiss

**NEW PRODUCTS EDITOR**  
Harv Weiner

**ART DIRECTOR**  
Lisa Ferry

**GRAPHIC ARTIST**  
Joseph Quinlan

**CONTRIBUTORS:**  
Jon Elson  
Tim McDonough  
Frank Kuechmann  
Pellervo Kaskinen

**PUBLISHER**  
Daniel Rodrigues

**PUBLISHER'S ASSISTANT**  
Susan McGill

**CIRCULATION COORDINATOR**  
Rose Mansella

**CIRCULATION ASSISTANT**  
Barbara Maleski

**CIRCULATION CONSULTANT**  
Gregory Spitzfaden

**BUSINESS MANAGER**  
Jeannette Walters

**ADVERTISING COORDINATOR**  
Dan Gorsky

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 \$43.35. All subscription orders payable in U.S. funds only, via international postal money order or check drawn on U.S. 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 08077.

Cover Illustration by Bob Schuchman  
PRINTED IN THE UNITED STATES

## HAJAR ASSOCIATES NATIONAL ADVERTISING REPRESENTATIVES

<b>NORTHEAST</b> <b>Debra Andersen</b> (617) 769-8950 Fax: (617) 769-8982	<b>SOUTHEAST</b> <b>Christa Collins</b> (305) 966-3939 Fax: (305) 985-8457	<b>WEST COAST</b> <b>Barbara Jones</b> <b>&amp; Shelley Rainey</b> (714) 540-3554 Fax: (714) 540-7103
<b>MID-ATLANTIC</b> <b>Barbara Best</b> (908) 741-7744 Fax: (908) 741-6823	<b>MIDWEST</b> <b>Nanette Traetow</b> (708) 789-3080 Fax: (708) 789-3082	

Circuit Cellar BBS—24 Hrs 300112001240019600114 4k bps, 8bits, no parity, 1 stop bit, (203) 871-1988; 2400/9600 bps Courier HST, (203) 871-0549

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

Circuit Cellar INK makes no warranties and assumes no responsibility or 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 © 1993 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.

# INSIDE ISSUE 34

12

## CLOSED CAPTIONING *with the* MOTOROLA 68HC05CC1

by Janice Benzel & Linda Reuter Nuckolls

26

## *Vector Approach Simplifies* THREE-DIMENSIONAL GRAPHICS

by Fred H. Carlin

32

## GRAPHICS LCD *Control for* *Embedded Applications*

by David Erickson

2

Editor's INK/Ken Davidson  
A Match Made in Heaven

6

New Product News  
edited by Harv Weiner

44

Firmware Furnace/Ed Nisley  
I/O Time: The '386SX Project Gains a Timer

54

From the Bench/Jeff Bachiochi  
Squeeze That Battery 'Till It's Dry

60

Silicon Update/Tom Cantrell  
Audio Rx-Skippy CDs? Tangled Tapes?  
Call an MD...

66

Embedded Techniques/John Dybowski  
Interchip Traffic

78

Patent Talk/Russ Reiss

85

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

96

Steve's Own INK/Steve Ciarcia  
The Club

81

Advertiser's Index

# NEW PRODUCT NEWS

Edited by Harv Weiner

## HIGH-PERFORMANCE APPLICATION COPROCESSOR

Innovative Integration has introduced a high-performance IBM PC/AT plug-in Application Coprocessor board. The PC31 combines the Texas Instruments TMS320C31 microcontroller and the ANSI C or the Forth programming languages to provide a complete integrated solution to the development and implementation of real-time applications and IBM PC front-end signal processing.

The TI320C31 32-bit architecture is optimized for computationally intensive floating-point digital signal processing algorithms and operates at a sustained 17-MIPS, 33-MFLOPS throughput. The processor provides two 32-bit counter/timers, seven prioritized interrupts, two synchronous serial ports, on-chip

RAM/ROM, and a 40-bit single-cycle floating-point multiplier/accumulator. Additionally, the full-length IBM plug-in card features an eight-channel prioritized interrupt controller, 2-Mbps DUART, six 8-bit digital I/O ports, three 16-bit counter/timers, two 16-bit 180-kHz ADCs (one of which is 8:1 multiplexed into a programmable-gain amp), and four 16-bit 180-kHz DACs. Up to 16 megawords of 0-7-wait-state RAM is supported. Peripheral cards addressing large data gathering and audio applications are available.

Software is developed and tested using the optimizing 100% ANSI-compatible C compiler or incremental Forth compiler. Applications are quickly developed and tested using the comprehensive integer and floating-point math, analog and digital I/O, and video display libraries provided with the Developers Package. All system and application procedures may conveniently be interactively edited, tested, and executed at full machine speed.

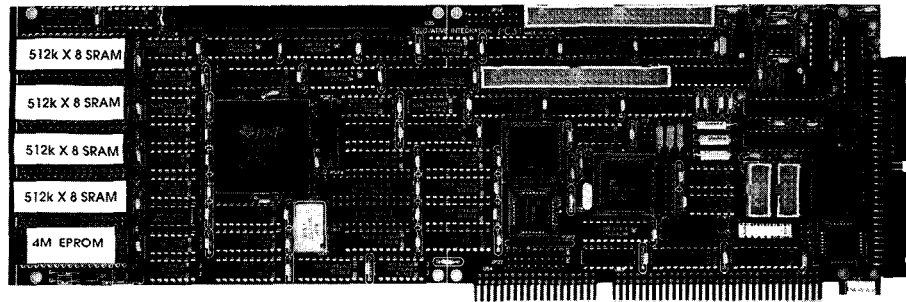
Applications for the board include real-time

control; PID; nonlinear control; audio digitizing, processing, and playback; digital filters; math coprocessing; and image processing.

The PC31 sells for \$850 in single quantities and complete development packages start at \$2995.

Innovative Integration  
4086 Little Hollow Pl.  
Moorpark, CA 93021  
(805) 529-7570  
Fax: (805) 529-7932

#500



## AC POWER SOURCE

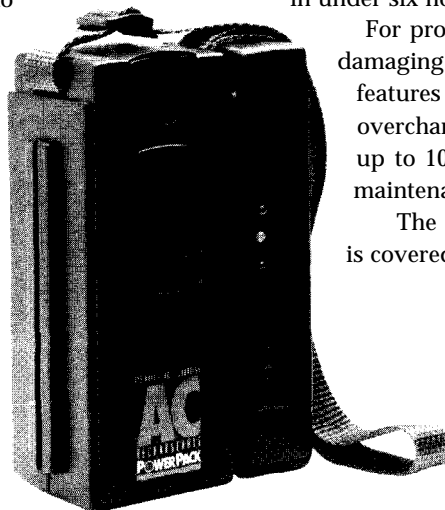
Innova Electronics has introduced the AC Power Pack, a portable, rechargeable power source specially designed for 120-volt AC/DC products up to 50 watts. The AC Power Pack provides an alternative to traditional portable power sources and combines both 12-volt DC and 120-volt AC capability. In addition, the Power Pack has a removable, stand-alone AC module that can be used to provide 120-volt AC power from any car cigarette lighter socket.

The entire unit measures 3" x 3" x 7.5", weighs only three pounds, and contains a 3.0-amp-hour battery. The AC inverter module measures 1.75" x 3" x 7.5" and weighs 0.75 pounds. The AC Power Pack is easily recharged in one to three hours through any vehicle cigarette

lighter while the engine is running or in four to six hours using the Innova AC Charger. Also available from Innova is a Solar Charger that will recharge the AC Power Pack in under six hours on a sunny day.

For protection against overloading and damaging circuits, the AC Power Pack features a built-in regulator that prevents overcharging. The unit can be recharged up to 1000 times on its replaceable, maintenance-free, sealed lead-acid battery.

The AC Power Pack sells for \$175 and is covered by a one-year warranty.



Innova Electronics Corp.  
17287 Mount Herrmann St.  
Fountain Valley, CA 92708  
(714) 432-1184  
Fax: (714) 432-7910

#501



# NEW PRODUCT NEWS

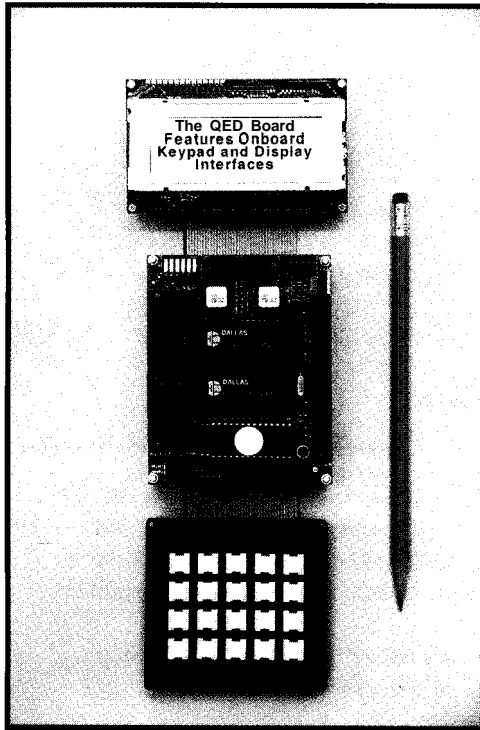
## INSTRUMENT DEVELOPMENT SYSTEM

The QED Product Design Kit from Mosaic Industries is a "generic instrument" that speeds development of instrumentation and control applications. Programmable from any PC or terminal, the kit integrates a 68HC 11-based controller board, LCD display, keypad, power supply, and prototyping board in an aluminum instrument enclosure.

The heart of the instrument kit is the QED Board, an embedded computer that hosts a resident high-level Forth programming environment in on-board ROM. Interactive debugging tools support breakpoint insertion, tracing, and single stepping. A built-in multitasking executive implements cooperative and time-sliced task switching, and on-board libraries include I/O device drivers as well as floating point, calibration, and matrix math functions.

The 3.2" x 4" board hosts up to 384K of memory including battery backed write-protectable RAM that eliminates the need for PROM burning while programming. Fabricated using double-sided surface-mount technology, the battery-operable board provides keypad and display interfaces, digital I/O, up to sixteen 8-bit and 12-bit ADC channels, up to eight 8-bit and 12-bit DAC channels, eight timer-controlled signals, and dual RS-232/485 serial ports.

An aluminum instrument enclosure houses all of the hardware. In addition to the installed QED board and prototyping board, there is room for two additional 3.2" x 4" circuit boards. Two 25-pin serial communications connectors and a power jack reside on the back panel



of the enclosure.

The display is a 4-line by 20-character high-contrast liquid crystal display with a visible display area of 1" x 3". The keypad features 4 rows by 5 columns and is 2.7" x 3". Custom labels can be inserted under the clear plastic key caps. A wall-mount DC power supply provides 6 VDC at up to 500 mA through a jack on the back panel of the enclosure.

The QED Product Design Kit is priced from \$875 and the QED board is priced from \$495.

Mosaic Industries, Inc.

5437 Central Ave., Ste. 1 • Newark, CA 94560 • (510) 790-1255 • Fax: (510) 790-0925

#502

## LOW-COST LED TESTER

A compact, hand-held unit for convenient testing, evaluation, and quality control checks of discrete LEDs has been introduced by Lumex Opto/Components Inc.

The LED Tester features single sockets with fixed current levels of 2, 5, 20, and 30 mA. Seven sockets at 10 mA are also provided so that similar LEDs can be easily compared for color and/or brightness. A single 9-volt battery powers the tester.

The LED Tester is 2 <sup>3</sup>/<sub>8</sub>" x 3 <sup>1</sup>/<sub>8</sub>" x 7 <sup>7</sup>/<sub>8</sub>" and weighs 3 ounces, complete with battery.

The LED Tester sells for \$38.00 including battery.



Lumex Opto/Components, Inc.

292 E. Hellen Road • Palatine, IL 60067

(708) 359-2790 • Fax: (708) 359-8904

#503

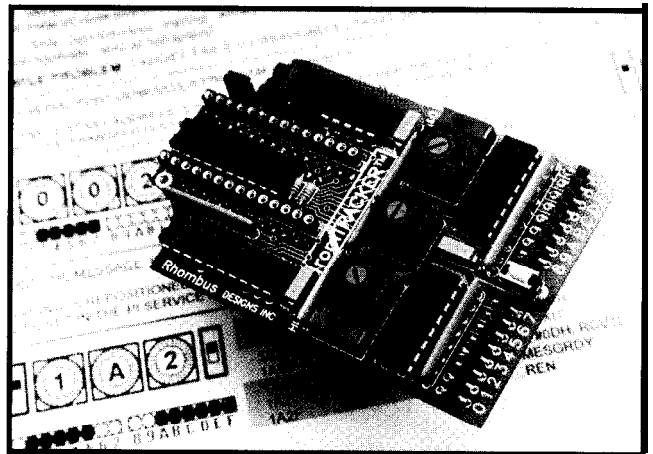
# NEW PRODUCT NEWS

## DEBUGGING TOOL FOR ROM-BASED SYSTEMS

Rhombus Design's romTRACKER debugging tool is designed specifically for ROM-based systems. It measures 2" x 3", is PLD based, and installs directly between the ROM and its socket. All of its functions are available while running out of ROM, and include tracing the execution of segments of code, locating the addresses of a hung program, and producing a hardware trigger on any ROM address.

The romTRACKER is self-contained with its own Start Trace address selection. The addresses that execute in the selected segment are stored and mapped onto an array of 16 LEDs. When operating at one address per LED, the resulting coverage of 16 consecutive addresses is quite adequate to show the details of how conditional code is executed. For each segment selected, correlating the address-mapped LEDs with the addresses in the program listing is quick and simple. Other modes allow expanded coverage. There are no cables to load or influence the target signals, it operates with any processor that uses a 28-pin DIP EPROM, and it takes its power (up to 50 mA) from the socket.

The universal operation and pocket size of romTRACKER will allow it to contribute to the user's debugging process regardless of the processor type and existing tools. Typical uses will include complementing



the operation of a ROM emulator, supporting the transition from RAM-based development to ROM production, use as the sole debug tool in combination with an EPROM programmer, and use as a portable tool on-site.

The romTRACKER sells for \$129 and is supplied with a shirt-pocket-size protective case.

Rhombus Designs, Inc.  
5 Woodlawn Green #220 • Charlotte, NC 28217  
(704) 525-3351

#504

## LOW-LEVEL MEASUREMENTS HANDBOOK

Keithley Instruments has published a 220-page handbook on making low-level measurements. The "Low-Level Measurements Handbook" is written both for the measurements expert and novice, with details on specific techniques for making even the most difficult and sensitive measurements. Illustrations and schematics accompany the measurements tutorials.

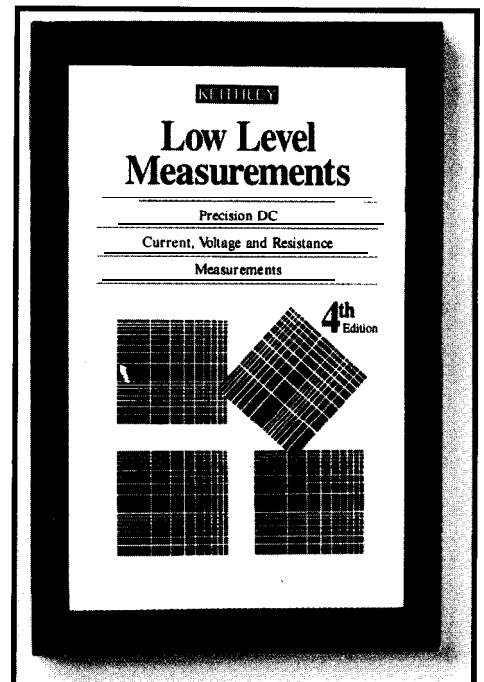
The updated handbook covers essential techniques for improving measurement accuracy, including sections on common low-level voltage, current, and resistance measurement techniques; typical applications; error sources and how to avoid them; and an instrument selector guide.

The handbook includes an introductory section that explains why special instrumentation is often required when making low-level measurements, along with specifications and circuitry to look for when buying instrumentation that will distinguish low-level from general-purpose instrumentation.

Step-by-step procedures and instructions are provided to the reader. Also included is a detailed glossary of terms.

For a free copy of the handbook, contact:

Keithley Instruments, Inc.  
28775 Aurora Rd.  
Cleveland, OH 44139  
(216) 248-0400  
Fax: (216) 248-6168

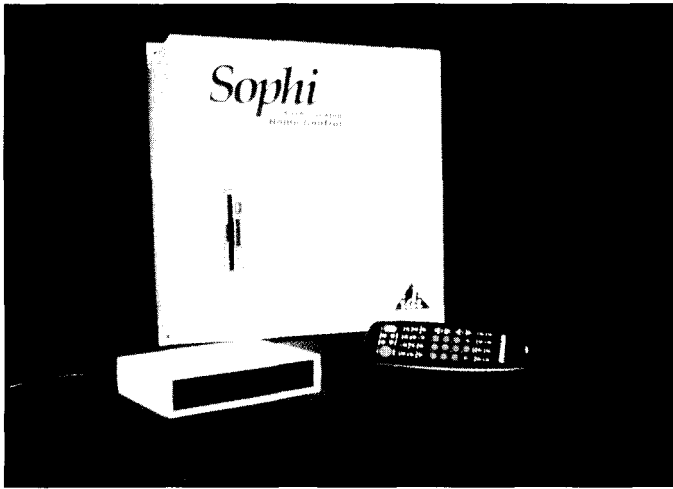


#505

# NEW PRODUCT NEWS

## HOME CONTROL SYSTEM

Digital Technology has introduced a Sophisticated Home Control System, **SOPHI**, that provides complete control of security, environment, lighting, appliances, and telephone functions. The system is easily installed in existing homes and extremely simple to operate. SOPHI is unique because it incorporates all of these features into one system.



SOPHI is connected to the existing cable TV wiring in a home and displays its status on an unused cable channel. It is controlled by a hand-held remote that also controls the TV, VCR, and audio system, or by any telephone in or out of the home. The TV screen displays a series of easily understood menus and the user is guided by a series of voice prompts.

The security system uses wired or wireless sensors, such as door and window sensors, fire and smoke sensors, and motion sensors. The unit can dial any telephone number and deliver prerecorded messages.

SOPHI provides multizone heating and air conditioning capability. The basic system will control up to four zones. Temperature changes are made with the hand-held remote through the TV screen.

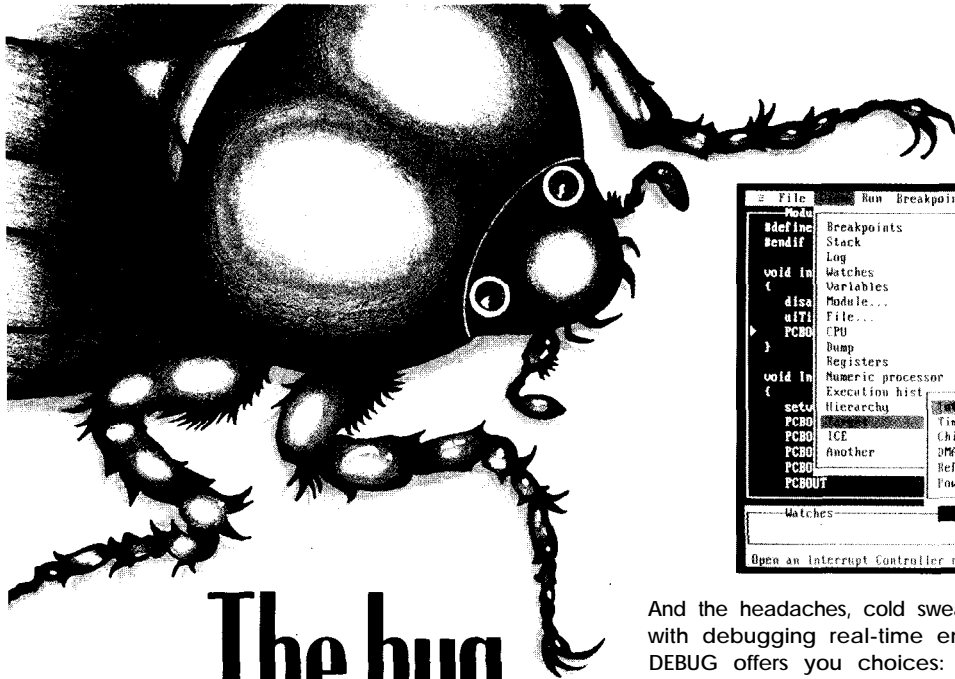
In addition, SOPHI includes a voice mail and call forwarding system and X-10 technology, which is used to control up to 256 lamps and electrical appliances.

Digital Technology, Inc.

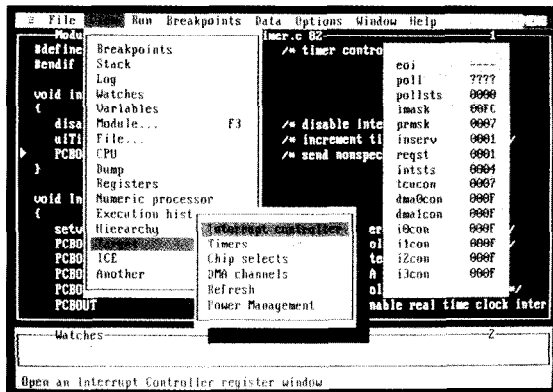
1000 Riverbend Blvd., Ste. R • St. Rose, LA 70087

(504) 467-466 • Fax: (504) 467-2146

#506



# The bug stops here



And the headaches, cold sweats and other symptoms associated with debugging real-time embedded applications. Paradigm DEBUG offers you choices:

- Intel or NEC microprocessors
- Remote target or in-circuit emulator support
- C, C++ and assembler debugging
- Borland, Microsoft and Intel compatibility.

Kickstart your embedded system with the only debugger family to have it all. Give us a call for Paradigm DEBUG ...before it's too late!

**800-537-5043**

## PARADIGM DEBUG

Proven Solutions for Embedded C/C++ Developers

PARADIGM: (607) 748-5966

FAX: (607) 748-5968

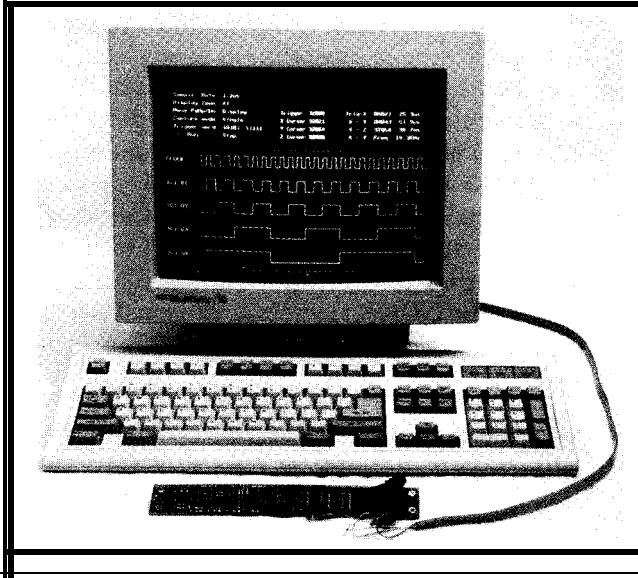
The Computer Applications Journal Issue #34 May 1993

**NOW SHIPPING!**  
C++ Templates, Clipboard, Breakpoint Groups & More!

# NEW PRODUCT NEWS

## SOFTWARE LOGIC ANALYZER

A new software program that converts an IBM-compatible computer into a fully functional logic analyzer has been introduced by Logixell Electronics. With the Real Logic Analyzer, up to five waveforms can be monitored through the standard PC parallel printer port. The user connects a circuit to the port by making a simple cable or by using Logixell's optional cable with universal test clips.



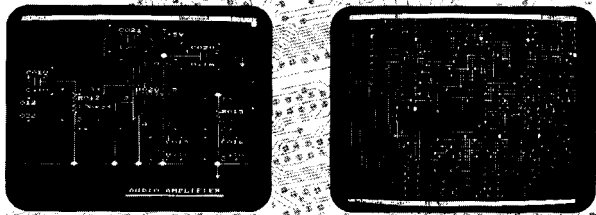
The software uses the computer's memory to capture 64K samples of data as fast as every 1.2  $\mu$ s (depending on computer). Full triggering capabilities allow for the trigger word to be set to any combination of High, Low, or Don't Care values. The trigger point may be set anywhere within the 64K buffer to allow for capturing of both pre- and post-trigger waveforms. Captured waveforms are displayed graphically on the computer screen and can be viewed at several different zoom levels. A continuous display mode allows for an oscilloscope-type of real-time display.

The Real Logic Analyzer comes with a 32-page instruction manual and program diskette for only \$35. The optional Test Cable sells for \$17.95.

Logixell Electronics

MI 1-2881 Richmond Rd. • Ottawa, Ont. K2B 8J5 Canada  
(613) 828-4159 • Fax: (613) 828-8954

## Schematic Capture to Error-free PCB



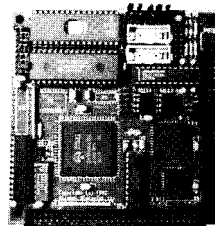
**EZ-LOGIC - \$99.00**  
**EZ-BOARD - \$149.00**

EZ-LOGIC and EZ-BOARD are "personal use" versions of and compatible with our professional level EZ-ROUTE PRO PCB Design System. These programs are perfect for design engineers who desire an economical yet powerful PCB design program. The complete system includes the DesignRule checker and a ability to view gerber plot files. The schematic capture module supports A through E size sheets, comes with user expandable library and outputs netlists. The PCB Layout module supports 8 layers, trace width from 0.001 inch to 0.255 inch, flexible grid, accepts netlists from several different formats such as Futurnet, PCAD, and EDIF. SMD component on both sides of the board and outputs to penplotters, gerber photoplotter, and dot matrix printers.

**For a FREE Evaluation call 1-800-972-3733**  
30 DAY NO-HASSLE MONEY BACK GUARANTEE

**ADVANCED MICROCOMPUTER SYSTEMS, INC.**  
**ams** 1460 S.W. 3rd St, Suite B-8, Pompano Beach, FL 33069  
(305) 784-0900 • FAX (305) 784-0904

**Finally A Proven Single Board Computer With More Power, Superior performance, That Meets Above Average Industry Standards, That Has Unequal Technical Support And, Oh, Yeah, It's Cheaper.**



- MMT-Z180**
- 2 serial I/O ports
  - 3 programmable parallel I/O ports
  - 2 32-pin configurable memory sockets
  - power fail detect interrupt and reset
  - counter-timers
  - watch dog timer
  - expansion connector

**Only \$159.00**

ALSO AVAILABLE: MMT-188EB, 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.

**Midwest Micro -Tek**

2308 East Sixth Street, Brookings, SD 57006  
(605) 697-8521, Fax (605) 697-8109



# NEW PRODUCT NEWS

## ARBITRARY WAVEFORM GENERATOR

Real Time Devices has announced the **FG102**, a low-cost PC-bus arbitrary waveform generator. The unique design of the FG102 uses the host computer's DMA and interrupt circuitry to transfer data, enabling it to generate extremely long waveforms. Two 12-bit analog waveforms can be generated simultaneously as a DMA-controlled background task. The waveforms can be generated by the mathematical functions in any high-level language or by an application program. Alternatively, the waveforms can be digitized real-world signals acquired from a data acquisition system and reproduced by the FG 102.

In contrast to RAM-based arbitrary waveform generators, the FG102's use of DMA and interrupts permits much larger files to be used. The waveform size is limited only by the amount of available hard disk space, since the FG102 uses a buffer swapping software technique to overcome DMA's 64K page barrier. This allows waveforms larger than 64K to be generated in real time.

The waveform data transfer rate is programmable; the maximum rate depends on the host computer. Data

transfer rates on an 80386-class machine are in excess of 100,000 points per second. The FG102 supports single and repetitive waveform generation and either DMA or non-DMA transfer modes. The buffered analog outputs have a 5- $\mu$ s settling time and four jumper-selectable output voltage ranges. The output waveform can be jumpered to an on-board audio amplifier to directly drive an 8-ohm speaker.

Applications for the FG102 include automated testing, acoustics, DSP, speech synthesis, or any application requiring computer generation of large arbitrary waveforms.

Software support for the FG102 includes programming examples in QuickBASIC, Turbo C, and Turbo Pascal. A diagnostic program is included to verify the FG102's operation and configuration.

The FG102 Arbitrary Waveform Generator sells for \$459.

Real Time Devices, Inc.  
820 N. University Dr. • P.O. Box 906  
State College, PA 16804-0906  
(814) 234-8087 • Fax: (814) 234-5218

#507

## BCC52 BASIC-52 COMPUTER/CONTROLLER

The BCC52 Computer/Controller is Micromint's hottest selling stand-alone single-board microcomputer. Its cost-effective architecture needs only a power supply and terminal to become a complete development or end-use system, programmable in BASIC or machine language. The BCC52 uses Micromint's 80C52-BASIC CMOS microprocessor which contains a ROM-resident 8K-byte floating-point BASIC-52 interpreter.

The BCC52 contains sockets for up to 48K bytes of RAM/EPROM, an "intelligent" 27641128 EPROM programmer, three parallel ports, a serial terminal port with auto baud rate selection, a serial printer port, and is bus-compatible with the full line of 30C-bus expansion boards. BASIC-52's full floating-point BASIC is fast and efficient enough for the most complicated tasks, while its cost-effective design allows it to be considered for many new areas of implementation. It can be used both for development and end-use applications.

### PROCESSOR

- 80C52-BASIC, 8-bit CMOS microcomputer
- \*jumper-selectable conversion to 80C31/80C32 functionality
- 8K bytes ROM (full BASIC Interpreter)
- 256 bytes RAM
- \*three 16-bit counter/timers
- 32 I/O lines
- 11 MHz system clock
- 6 interrupts

### MEMORY

- expandable to 62K bytes
- \*five on-board sockets
- up to four 6264 (8Kx8) static RAM
- either an 8K 2764 or 16K 27126 EPROM

B C C 5 2 BASIC-52 Controller/Board with 8K RAM  
**BCC52C** Lower-power all-CMOS version of the BCC52  
**BCC52I** Full industrial temperature range  
 BCC 5 2CX CMOS Expanded BCC52 w/32K RAM

### Input/Output

- \*console I/O RS-232 serial port
- \*line printer AS-232 serial port
- \*three 8-bit programmable TTL-compatible parallel I/O ports using a 8255 PPI
- \*alternate console RS-422/RS-485

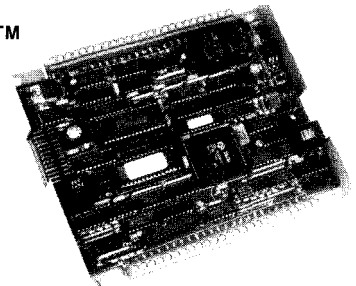
**To Order Call**  
**1-800-635-3355**  
 Tel: (203) 871-6170  
 Fax: (203) 872-2204  
 TELEX: 643331

	Single Qty.	100 Qty.
BCC52C	\$189.00	\$149.00
BCC52I	\$199.00	\$159.00
BCC 5 2CX	\$294.00	\$220.00
BCC 5 2CX	\$529.00	\$159.00

**MICROMINT, INC.** 4 Park Street, Vernon, CT 06066

#119

## C-PLC™ \$289!



- New C Programmable miniature controller
- Seven 1 O-bit analog inputs
- Seven digital inputs
- 1 O-bit DAC: voltage or current output
- Twelve digital/relay driver outputs
- RS-232/RS-485 serial ports
- Enclosure with LCD/Keypad available
- Expansion bus for additional, low cost I/O
- Easy to use **Dynamic C™** development software only \$195!

## Z-World Engineering

1724 Picasso Ave., Davis, CA 95616  
**(916) 757-3737** Fax: (916) 753-5141  
 24 hr. Information Service: (916) 753-0618  
 (Call from your **fax** and request data sheet #24)

#106

## FEATURES

12

Closed Captioning With  
The Motorola 68HC05CC1

26

Vector Approach Simplifies  
3-Dimensional Graphics

32

Graphics LCD Control For  
Embedded Applications

# Closed Captioning With The Motorola 68HC05CC1

By July 1, 1993, all televisions sold with screens 13" or larger must be able to decode and display closed captions. The 68HC05CC1 processor makes adding such capability to existing circuits much easier.

## FEATURE ARTICLE

**Janice Benzel &  
Linda Reuter Nuckolls**

**O**he next TV you purchase will most likely have built-in closed-caption capability, thanks to the Television Decoder Circuitry Act of 1990. Congress enacted this law to eliminate the need for consumers to purchase expensive add-on decoder hardware to view the captions that are included in television broadcast signals. By July 1, 1993, televisions with screens 13" (or greater) in size sold in the U.S. must be able to decode and display closed captions.

The benefits of this law will be widespread. For 23 million hearing impaired Americans, the Television Decoder Circuitry Act means increased access to captioning at home and in public facilities. In addition, extended data services, to be transmitted in the caption format, will provide new features with universal appeal. These services could include program information or captions in a second language or at another reading level.

The Decoder Act forced television manufacturers into a race to define a closed-caption standard and incorporate it into the 6-billion-dollar U.S. TV market. The manufacturers have kept one eye on the calendar and the other focused on cost issues. Semiconductor suppliers have hastily designed parts for the closed-caption niche, but most of the multichip sets or specialized ICs they offer require external controllers. The need for multiple ICs increases system complexity and cost.

### A SINGLE-CHIP SOLUTION

The Motorola 68HC05CC1 is a true single-chip solution, since it

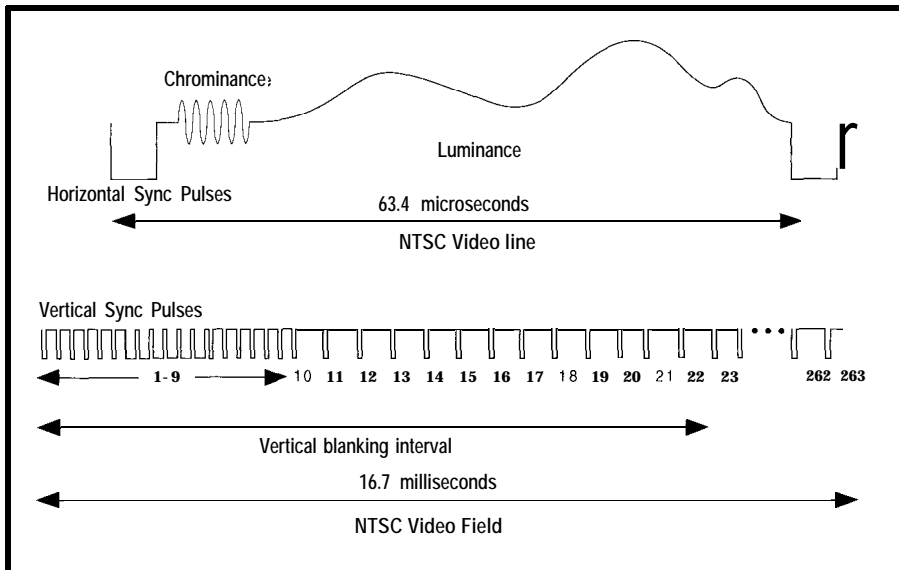


Figure 1--Each scan *line* of an NTSC picture *starts* with a horizontal sync pulse and *lasts* 63.4 microseconds. Each field, made up of 262.5 scan lines, starts with an invisible vertical blanking interval. Caption data is embedded in line 21 of the VBI.

combines the controller and caption systems on one die. The CC1, an offspring of the Motorola "T" series of TV microcontrollers, was developed by Motorola in partnership with Thomson Consumer Electronics. The 6805 based microcontroller includes peripherals used for basic TV control with special hardware and software for the decoding and display of closed captions.

The CC1 integrated circuitry for data extraction and special display functions for closed captioning, are what makes the CC1 a new generation part. Photo 1 shows the various modules of the CC1 die. The closed-caption hardware on the CC1 consists of a Data Slicer module (DSL) and an On-Screen Display module (OSD) which are driven by an on-chip Phase

Locked Loop (PLL). These modules provide the data extraction and caption display functions, respectively, for closed captioning.

The CC 1 includes several other peripherals on-chip to perform other tasks not related to captioning. The Pulse Width Modulator is used for audio and video control. Serial communication with external devices is provided with the Synchronous Serial Interface. The Pulse Accumulator performs input pulse measurements, or pulse counting, for remote control interpretation. For communication from the TV chassis back to the controller, there is a 5-bit Analog-to-Digital Converter and Comparator system. The functions performed by these peripherals are designed not to burden the 68HC05CC1 CPU, allow-

ing time for the more CPU-intensive closed-caption decoding tasks.

We'll show you how we've used the 68HC05CC1 to view captions. We'll set the stage by delving into the rules for closed captioning and by exploring the interior of the CC 1. We're going to describe the software and hardware used in our setup, so you'll understand closed-caption programming challenges and how the CC1 can help you meet them.

## CLOSED-CAPTION REQUIREMENTS

The Federal Communications Commission (FCC) was empowered by the TV Decoder Circuitry Act to establish rules dictating performance and display standards for caption decoders. The FCC worked from a proposal developed by a task force from the Electronic Industries Association (EIA). The EIA, in turn, referred to the captioning precedents set in the 1980s by the Public Broadcasting Service (PBS) and the National Captioning Institute (NCI). In order to maintain compatibility with existing decoders, the PBS/NCI data transmission format was left intact, but modifications were made to the display methods to improve captioning and allow manufacturers additional flexibility. On April 12, 1991, the FCC adopted the Report and Order describing caption requirements.

To understand how captioned data is transmitted, it's best to start with the "big picture" of NTSC video-the U.S. standard-shown in Figure 1. Video signals contain timing, chromi-

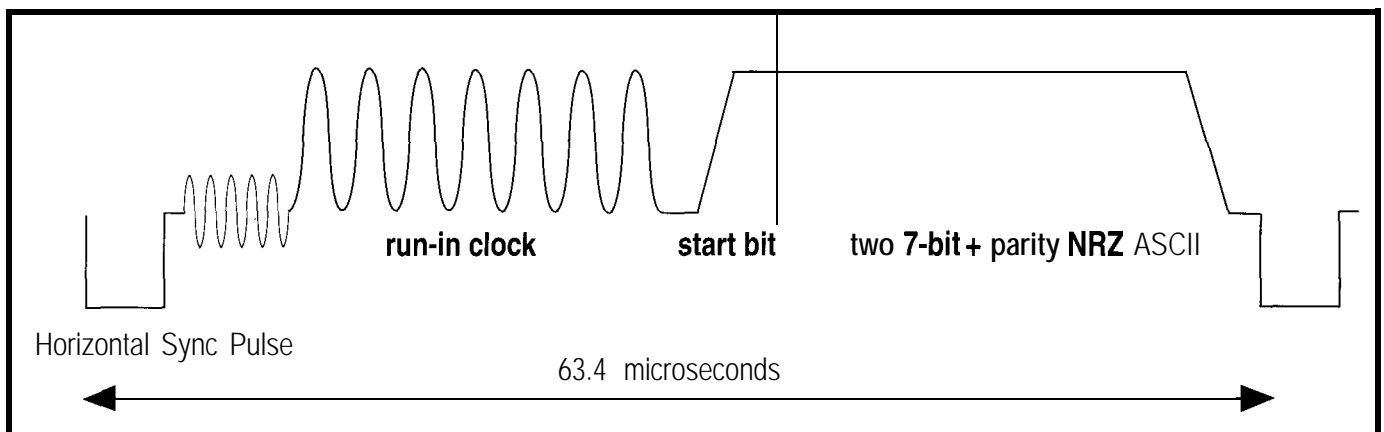


Figure 2-Caption data is found in line 21 of the vertical blanking interval. The run-in clock provides a timing and amplitude reference for the start bit and is followed by 16 bits of data. The effective data rate is 60 bytes per second.

nance (color), and luminance (brightness) information. The basic component of a video signal is the scan line, which begins with a horizontal sync pulse and represents one horizontal pass across the screen. A series of 262.5 lines constitutes a field, or one vertical pass over the screen. Vertical sync pulses are contained in the invisible vertical blanking interval, which occurs at the start of a field and has a duration of several scan lines. Two fields are interlaced to form a complete picture, or frame, which consists of 525 lines. The frame rate is approximately 30 Hz, so the field (vertical) rate is twice that at 60 Hz, generating a line (horizontal) frequency of 15.75 kHz.

Caption data is embedded in line 21, which occurs during the vertical blanking interval. Field 1 was established by PBS/NCI as the location for caption data, and field 2 has been recommended to carry extended services information. The format for line 21 is shown in Figure 2. A horizontal sync pulse indicates a new line, followed by a run-in clock which provides a timing and amplitude reference for the start bit and 16 bits of NRZ data to follow. Each field supplies two data bytes of seven bits plus parity. The data rate for closed captioning on one field is 60 bytes per second, or 480 bits per second.

Caption data is processed as received, and each byte pair represents a double-byte control code or two single-byte visible characters. There are 96 ASCII visible characters, and four categories of control codes: special characters, mid-row codes, preamble address codes, and miscellaneous control codes. The special characters are visible non-ASCII characters such as the music note; there are 16 special characters, bringing the total printing character set to 112. The other three types of control codes configure the display; before we go into detail on them, a discussion of caption formats is in order.

## CAPTION DISPLAY

The FCC allows two display modes-caption and text-although text is optional and is only briefly

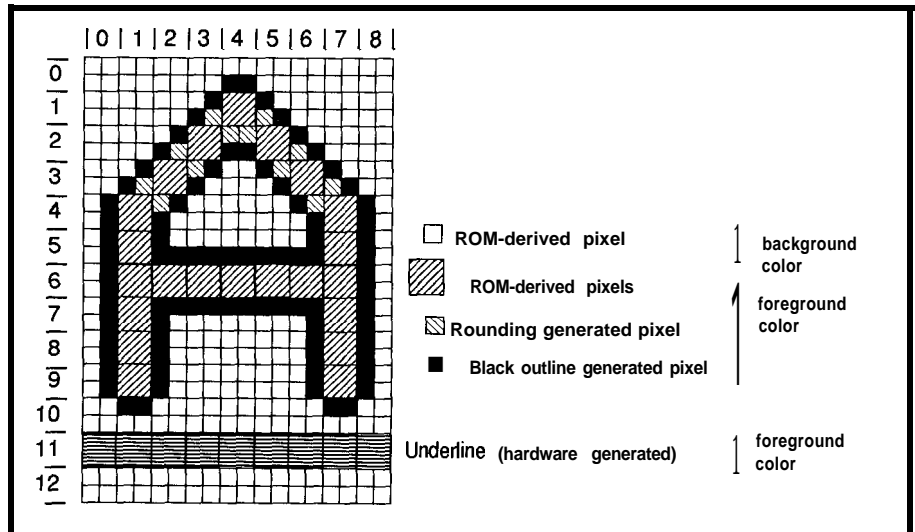


Figure 3—The OSD ROM character has rounding and outline attributes added to it in addition to underline, italics, and color information.

described. Both modes use a 15-row by 32-character display area. Caption mode consists of a maximum of four rows of characters displayed in one of three styles: pop-on, roll-up, or paint-on. The features which distinguish the three styles are how they address the screen and how they use memory.

Pop-on style uses two 4x32 byte memory banks: one for displayed data and the other for undisplayed data. A caption is stored in undisplayed memory as it is received. When the caption is complete, memories are swapped, and the displayed memory becomes undisplayed and vice versa. If you have seen captioning, you are probably familiar with this style.

Roll-up style uses only one memory bank, since the caption data is displayed as soon as it is received. What makes roll-up style unique is the way in which it addresses the screen. The first row displayed establishes a base row where each subsequent row will appear. When a carriage return is received, the top row disappears, the remaining rows soft scroll upward one row, and the new row is entered at the base row location. Roll-up style can be displayed with window sizes of two, three, or four contiguous rows. This is the most popular style for captioning news programs.

Paint-on style is similar to roll-up in that captions are displayed immediately, and only one memory bank is used. Paint-on style rows do not need to be contiguous and do not scroll.

Characters are swept across the screen as if applied with a brush. Paint-on is the least frequently used style.

Text mode is the optional display mode which can show up to 15 rows of 32 characters simultaneously. It addresses the screen like the roll-up style of caption mode.

Captioned text in any mode or style may take on several attributes, such as: color, italics, underline, and flash. Provisions are made for foreground color selection from a palette of seven colors, but compliance is optional. Although not considered an attribute, the background window surrounding the caption must have color menu choices of black and transparent. Italics are required, but may be implemented as a special character set, or by slanting the standard character set. The underline attribute adds a line under printing characters, and the flash attribute causes text to blink on and off. Attributes are assigned by preamble address codes, mid-row codes, and miscellaneous control codes.

## CONTROL CODES

Preamble address codes are used at the beginning of a row to identify the row number (location on the screen), an optional indentation, and the default attributes (color, italics, and underline) for the row. Preamble address codes are nonspacing characters, though they may alter the character cursor to indent. Indentation is

nondestructive, so a row that already has characters in it may be modified.

Midrow codes are used, as the name implies, in the middle of a row of characters. Midrow codes affect color, underline, and italics. They can also disable the flash attribute. Note that a midrow code appears as a background space, limiting the use of attributes to emphasize entire words or phrases.

Miscellaneous control codes perform all of the odd jobs of captioning. The most complicated miscellaneous codes are those that initiate, continue, or change display modes or styles. They are sent frequently, even when the style isn't changing, to ensure that switching receiver channels does not disrupt captioning. There are also miscellaneous codes which affect the memory banks, by either erasing displayed or undisplayed memory, or swapping memories. The Carriage Return code causes the display to soft scroll upward in caption mode's roll-up style or in text mode. Other miscellaneous

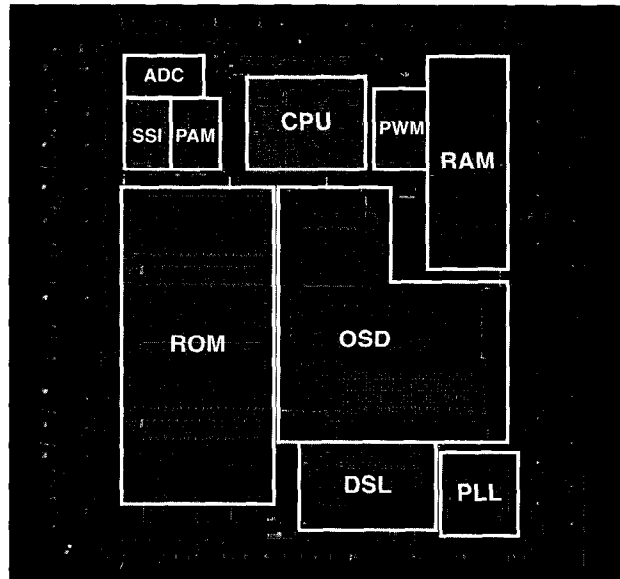


Photo 1—Close to half of the 68HC05CC1 die is dedicated to closed caption decoding and control. The rest contains the normal CPU, ROM, RAM, and support sections.

commands can modify the character cursor or row contents by backspacing, tabbing, or deleting to the end of the row. Flash is the only attribute set by a miscellaneous code; it affects all subsequent characters in a row until the next midrow code.

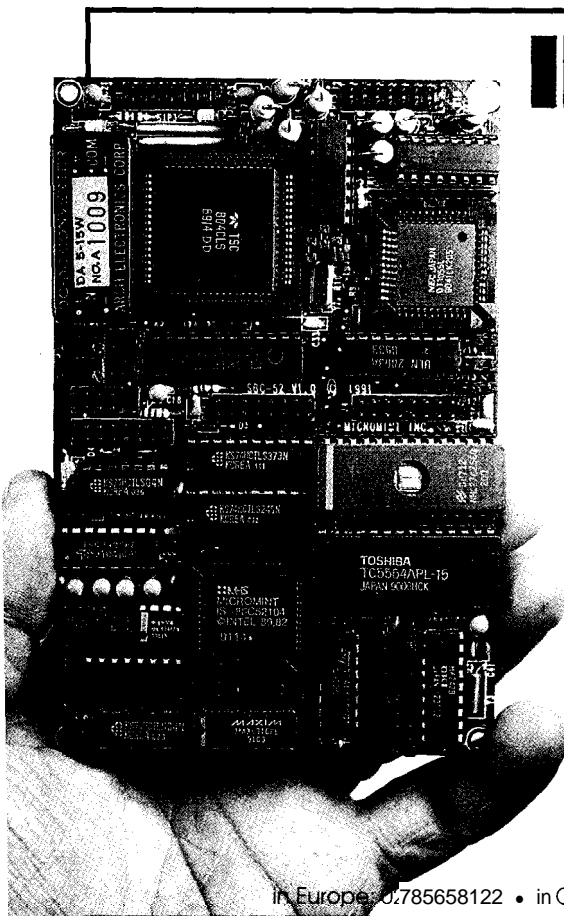
Now you should have a good feel for how captions must be decoded and

displayed. It would be impossible for us to relate all of the requirements and recommendations made by the groups involved in defining closed captioning. There are several documents which you can obtain for

more information. A list of these references is included at the end of this article.

### THE 68HC05CC1 CAPTION SUBSYSTEM

The DSL and OSD perform the decoding and display tasks of the caption subsystem, but they could not



# BRUTE-52

**Maximum Power  
Minimum Space**

Micromint's BRUTE-52 is the ultimate compact controller. One look at the list of features will tell you that this full-featured controller has the power to crush your most demanding applications:

- CMOS 80C52/80C32
- Three 16-bit counter timers
- 11.0592 MHz System Clock
- Hardware Watchdog Timer
- Hardware Clock-calendar
- Optoisolated Serial Communications RS-232 or RS-485/300-9600 bps'
- Optoisolated Serial Printer Port, RS-232 150-9600 bps
- 5V-only Operation
- Up to 56 Kbytes RAM and/or EPROM
- 1 Kbit EEPROM
- 12-bit parallel TTL/O
- 8-bits buffered high-voltage, high-current outputs
- 8-bits optoisolated non-polarized DC inputs
- 12-bit plus sign analog-to-digital converter 8 channels! 60 Samples/second! 12 mV resolution!
- 12-bit digital-to-analog converter 2 channels! 1.2mV resolution! Selectable ranges!
- Only 3.5 x 5.3 Inches!
- operates at 0-70°C
- Consumes only 100-200mA (depending on configuration)
- Use networked or stand-alone
- DEMO/Diagnostic ROM

BRUTE-52 offers you *all* these features at only \$459! (\$379/100 quantity OEM) We also have a starter system for \$289. When you add in Micromint's renowned quality, services, and support, you won't find a better value in compact control.

To order BRUTE-52, or for more information, contact:

**Micromint, Inc.**

4 Park Street • Vernon, CT 06066  
Phone (203) 871-6170 • FAX (203) 872-2204

In Europe: 0785658122 • in Canada: (514) 336-9426 • in Australia: (02) 888-6401 • Distributor Inquiries Invited!



function without the PLL. The OSD and DSL both need to be synchronized with the TV chassis, and the OSD requires a fast clock for dot processing. The PLL provides several stable frequencies which are phase locked to the horizontal sync from the chassis. The fastest of these is a 28.2-MHz clock which provides the internal horizontal timing for the character pixel display. There are two pins on the CC1 that are dedicated to tailoring the loop filter and center frequency.

## DATA SLICER

The Data Slicer consists of circuitry for sync and data recognition, line detection, and data acquisition. Sync and data information from the composite video pin is separated by the slicer circuitry. The separated sync contains both vertical and horizontal sync pulses. The difference in time between the rising edge of the horizontal sync and the rising edge of the vertical sync distinguishes between fields; data can be extracted from both fields.

A line detector counts horizontal lines from the vertical sync interval to find line 21, which contains the closed-caption information. This triggers the data acquisition block to begin sampling data. Sampled data is converted from serial to parallel format and stored in registers along with the results of the parity checking.

## ON-SCREEN DISPLAY

The On-Screen Display consists of a register array, character ROM, synchronization, and output blocks. Patterns for characters that are referenced in the register array are fetched from the OSD character ROM and processed by the output block under control of the synchronization logic.

The register array contains 9 bytes for display control, position, status and test. In addition, there are 34 character registers which hold the OSD ROM page addresses of the characters to be displayed on a single row. The character registers and some of the control registers are double buffered, which allows for dual OSD/CPU access. The OSD can be enabled after the character

Listing 1—Midrow code attributes are deciphered from the second byte, combined into an OSD video control code, and inserted into the RAM caption buffer.

```

* MRPROC: Mid Row control code PRoCessing
*           Interprets a Mid Row control code.
* Inputs: none
* Outputs: none
* Flags: none

MRPROC LDA  CURMODE      see if ctrl code is for selected mode
        BEQ  MRDONE      ignore ctrl code if mode not set yet
        CMP  SELMODE
        BNE  MRDONE      ignore codes not for selected mode
        LDA  #CHMRC      update previous character indicator
        STA  PREVCH
        JSR  GETATTR      get color, italics, & underline attr
        JSR  PUTVC        put character in memory
        LDA  VISCHAR      retrieve MKC that was put in memory
        STA  CCVCTL1      update current video control #1 value
MRDONE RTS

-----

* GETATTR: GET ATTRIBUTES from control code
*           Parses control code for color, italics, and underline
*           information.
* Inputs: none
* Outputs: attributes in video control #1 format in A
* Flags: none

GETATTR LDA  CHBUF2      ; get attr (2nd) byte of control code
        LSRA              ; shift to get rgb as bits 2-0
        AND  #COLORS     ; mask off all other bits
        BEQ  WHITE       ; 000 = white
        CMP  #COLORS
        BNE  GETUL
ITALICS LDX  PREVCH      ; check to see if this is PAC or MRC
        CPX  #CHPAC
        BEQ  GETIT       ; if PAC, italics makes color white
        LDA  CCVCTL1     ; if MRC, italics uses previous color
        AND  #COLORS     ; get current video ctrl #1 color value
GETIT   ORA  #OSDITMK    ; set italics bit
        BRA  GETUL
WHITE   ORA  #COLORS     ; set r,b,g = 1 for white
GETUL   BRCLR CCUL,CHBUF2,NOUL
        ORA  #OSDULMK    ; set underline bit if indicated
NOUL    ORA  #VC1ADD     ; add two MSBs to indicate video ctrl #1
        RTS

```

and default control codes for the first row to be displayed are loaded into the registers.

The OSD character ROM contains 128 9x13 bit matrices that can store a custom character set. Individual bits in each character matrix block represent a square of four video pixels of either foreground or background video. The contents of the OSD ROM are specified at the time of product order along with the contents of the program ROM.

The synchronization block in the OSD controls the timing of the display of closed-caption data. It uses the

hsync and vsync external inputs from the chassis and the PLL output frequencies as references. The sync block maintains timing for dots, characters, lines, and fields. The event line number, stored in an OSD register, indicates the position of the first scan line of a character row to be displayed. When a match occurs between the scan line number and the event line, the display process begins.

Character codes are sequentially read from the character registers during each scan line. Each code, along with the scan line number within the character row, is used as an address to

fetch the appropriate 9-bit "slice" of horizontal data from the character ROM. Vertically adjacent line patterns are also fetched to provide data that the output logic uses to interpolate the character cells for rounding and black outline of characters. A sample OSD ROM character with added rounding and outline attributes is shown in Figure 3. The output logic also adds underline, italics, and color information. The bit patterns drive the red, green, and blue color outputs and fast luminance blanking output which communicate with the color guns of the TV.

At the time of an event match, the user is signaled with an interrupt that indicates the OSD is ready for the next row of character codes to be loaded into the OSD registers. The shortest time available to the program for updating the registers with the next row codes extends from the beginning of the currently displayed character row (at the time of the interrupt) to the end of that row. This is the case if the next row to be displayed is directly below the currently displayed row. A rate of 262.5 scan lines (one field) in 1/60 of a second gives 63.49 microseconds per line. Each character row has 13 lines, so that leaves a minimum interval of 825 microseconds to load the next row codes. The 4-MHz internal bus frequency of the CC1 is double that of its predecessors in order to facilitate servicing of the OSD.

## OSD CLOSED-CAPTION FEATURES

The architecture of the OSD provides the flexibility required to perform closed captioning, but the display aesthetics are determined by the received data. Video control codes, which appear on the screen as a background color space, can be inserted anywhere in the character row. Video control codes are used to implement the closed-caption mid-row codes, which can change the current character foreground color, italics, and underline selections. The closed-caption miscellaneous control code for character flash uses the video control codes as well. The ability to change the appearance of text within a row is

**Listing 2—CPDATA** copies the RAM display buffer and the video control code, matrix register, and event line to the OSD register block.

```

*CPDATA: CoPy DATA from display buffer
* Copy the video control #1 character and character
* data from the display memory row into the OSD.
* Inputs: Index to row to be copied in X
* Outputs: none
* Flags: none

CPDATA LDA VC1BUF,X      copy video control byte #1
      STA OSDVC1
      LDA ELBUF,X      copy event line
      ADD RUPOFF      add Roll Up offset
      STA OSDELN
      LDA CCVCTL2      copy video control byte #2
      STA OSDVC2
      LDA CCHD         copy horizontal delay
      STA OSDHD
      LDA CCMR         copy matrix range
      STA OSDMR
      LDA CCBOR        copy border
      STA OSDBOR
      LDA #MAXCH      determine index into display buffer
      MUL
      TAX
      STX TBUFPTR     offset from start of buffer to char
      LDX #OSDCH1
      INCX
CPLoop STX TOSDPTR     update pointer to OSD registers
      LDX TBUFPTR     get pointer to buffer
      LDA OSDBUF,X    get character from buffer
      INCX
      STX TBUFPTR     update buffer pointer
      LDX TOSDPTR     get pointer to OSD
      STA             copy character to OSD
      INCX
      CPX #OSDCH34    copy 32 bytes
      BNE CPLoop
      CLR OSDCH1      clear first and last characters in row
      CLR OSDCH34
      LDA DOSOLID     first & last solid spaces selected?
      BEQ CPDONE      don't add solid spaces if not selected
      BRCLR OSDBKS,OSDVC2,CPDONE
      LDX #OSDCH1    ; search for first solid space
SPFIRST
      INCX            ; look for visible chars/MRCs in row
      CPX #OSDCH34  ; only search through 32 characters
      BEQ CPDONE
      LDA ,X
      BEQ SPFIRST    ; if a nonborder space character
      DECX
      LDA #SPACE     ; then put a space in previous location
      BRCLR OSDUL,OSDVC1,NOVC1UL
      LDA OSDVC1     ; if VC1 sets underline, delay
      BCLR OSDUL,OSDVC1; underline set to 1st space
NOVC1UL
      STA ,X         ; store 1st "space" (space or new VC1)
      LDX #OSDCH34  ; search backward to last solid space
SPLAST DECX         ; look for visible chars/MRCs in row
      CPX #OSDCH1   ; only search through 32 characters
      BEQ CPDONE
      LDA ,X
      BEQ SPLAST    ; if a nonborder space character
      INCX
      LDA #DEFVC1   ; then out a default VC1
      STA ,X        ; (to remove underline) in last space
CPDONE
      RTS

```

very important to a hearing-impaired individual watching a captioned broadcast, since it adds another dimension of expression to an otherwise lifeless line of text.

The roll-up closed-caption style requires that the display scroll vertically in a window. The OSD includes a special control register which, when modified in conjunction with the event line register, results in soft scrolling. The upper and lower nybbles of the matrix range register are used to select the scan lines where character foreground should begin and end, respectively. A character row consists of 13 scan lines; any lines outside the character matrix range appear as background. Scrolling a roll-up caption involves shifting the existing rows up by decreasing their event lines, and appending the new row by decreasing its event line while increasing its matrix range. Of course, the changing of the event line and matrix register must occur at an appropriate frequency. The FCC recommends a rate of one line per frame.

## A CLOSED-CAPTION APPLICATION

Our goal was to program the 68HC05CC1 to decode and display closed captions in a television. We did not incorporate the code required to control general TV functions and interact with the user. This simplified the programming task, but we wrote the software to accommodate additional burdens on the CPU, RAM, and ROM.

The critical issues in software development were time and memory. The time constraints were set by the service requirements of the OSD. Memory was given special consideration because of the large amount of RAM required to store the captions. The 16K ROM was more than sufficient for the 2K of decoding software.

The OSD can be very demanding. The minimum interrupt interval of 825 microseconds makes OSD underflow, and subsequent display distortion, a distinct possibility. We prevented underflow by minimizing the OSD service routine and by using SEI instructions sparingly throughout

the rest of the code. The only code segments protected from interruption are those that modify the buffer containing the screen addresses of the caption rows.

The concern with RAM requirements was that text mode requires nearly 512 bytes, and the CCI, not intended for full text mode, has only 544 bytes. We made a compromise that limited the text mode window to eight rows, and shared the text memory bank with the memory used for caption mode.

## SOFTWARE FUNCTIONALITY

The software has two main tasks: interaction with the CCI display subsystem, and closed-caption data interpretation. The interaction with the DSL and the data interpretation occur serially, in parallel with the interaction with the OSD.

The data slicer itself doesn't require much assistance. After initializing the control registers, all that needs to be done is a read of the data registers and status register upon an interrupt.

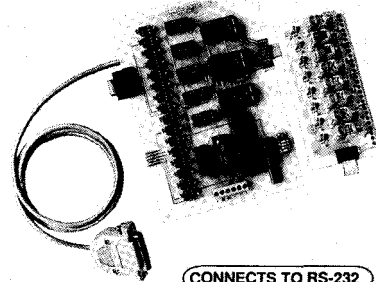
Once the two bytes of caption information have been retrieved from the data slicer, the interpretation phase begins. The first step is to determine whether the two bytes form a control code or two visible characters; a data parity check is also made for either case. Correct visible characters are copied directly into the RAM caption buffer, since there is a one-to-one mapping from ASCII code to OSD character address. Control codes may also end up as characters in the caption buffer, or they may modify the buffer, its pointers, or the actual display.

Control code processing includes checks for duplication and data stream selection. The leading byte of a control code addresses one of two data streams chosen by the viewer. All subsequent data, until the next control code, is directed to the data stream indicated in the first byte, and the software must discard the unused control codes and visible characters.

Both bytes of a control code are used to determine its type, and each type is processed separately. Once a

# RELAY INTERFACE

PROVIDES SOFTWARE CONTROL OF RELAYS

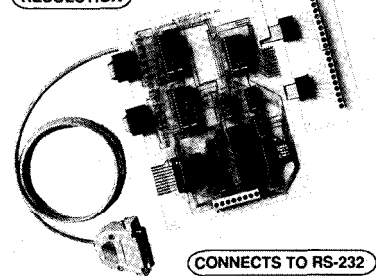


CONNECTS TO RS-232

**AR-16 RELAY INTERFACE** ..... \$89.95  
Two 8 channel relay output ports are provided for control of up to 16 relays (expandable to 128 relays using EX-16 expansion cards). Each relay output port connects to a relay card or terminal block. A variety of relay cards and relays are stocked. Call for more info. RS-422 available (distances to 4,000 feet). PS-4 port selector may be used to control satellite AR-16 interfaces. (up to 16,384 relays)  
**RD-8 REED RELAY CARD (6 relays, 10 VA)** ..... \$49.95  
**RH-8 RELAY CARD (10 amp SPDT 277 VAC)** ..... \$69.95  
**EX-16 RELAY EXPANSION CARD (16 channel)** ..... \$59.95

# ANALOG TO DIGITAL

8.10 & 12 BIT RESOLUTION



CONNECTS TO RS-232

**ADC-16 A/D CONVERTER (16 channel, 6 bit)** ..... \$99.95  
input temperature, voltage, amperage, pressure, energy usage, energy demand, light levels, joystick movement and a wide variety of other types of analog signals. Inputs may be expanded to 32 analog or 128 status inputs using the AD-16 or ST-32 expansion cards. 112 relays may be controlled using EX-16 expansion cards. Analog inputs may be configured for temperature input using the TE-6 temperature input conversion. RS-422 available. PS-4 port selector may be used to connect satellite ADC-16 interfaces (up to 4,096 analog inputs/16,384 status inputs and 14,336 relays). Call for info on 10 & 12 bit converters. (terminal block and cable sold separately)  
**ST-32 STATUS EXPANSION CARD** ..... \$79.95  
input on/off status of relays, switches, HVAC equipment, thermostats, security devices, smoke detectors and other devices including keypads and binary coded outputs. Provides 32 status inputs (opto isolators sold separately)  
**TE-6 TEMPERATURE INPUT CONVERSION** ..... \$49.95  
Includes 6 temperature sensors & terminal block. Temperature range is minus 40 to 145 degrees F.  
**PS-4 PORT SELECTOR (4 channels RS-422)** ..... \$79.95  
Converts an RS-232 port into 4 selectable RS-422 ports.  
**TOUCH TONE DECODER** and other serial interfacing products available. Call for free information packet.

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

Use our 600 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) 464-9656  
Use for information, technical support & orders  
ELECTRONIC ENERGY CONTROL, INC.  
380 South Fifth Street, Suite 604  
Columbus, Ohio 43215

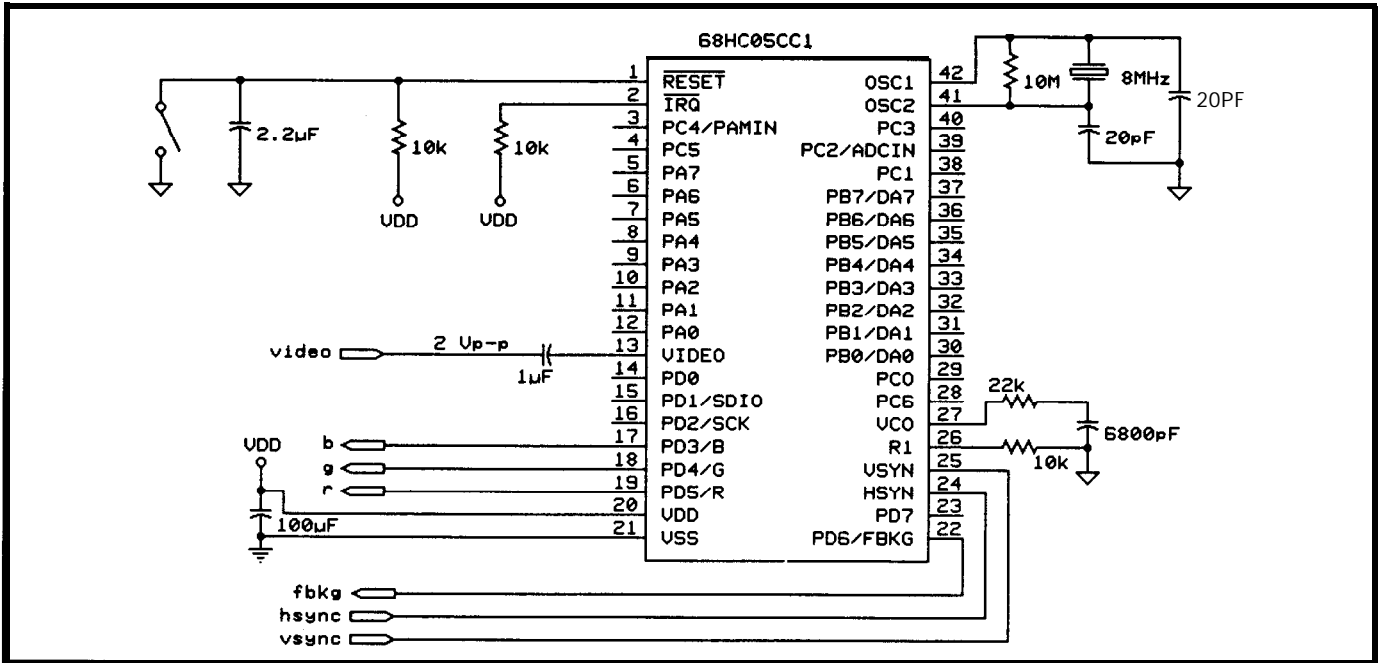
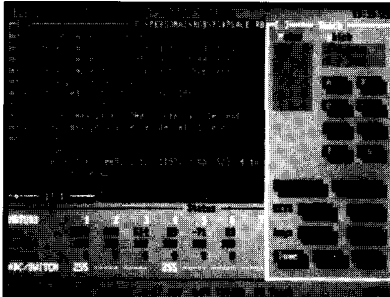


Figure 4—Using a minimum of external components, the MC68HC05CC1 accepts video, horizontal sync, and vertical sync, and directly drives the red, green, and blue CRT guns in the television, superimposing closed captions when necessary.

special character has been identified, its second byte is translated into the appropriate OSD character code and treated as a visible character. Midrow code attributes are deciphered from

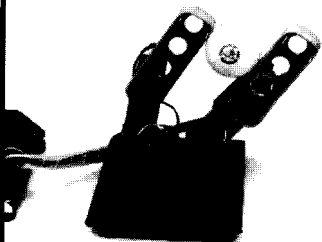
the second byte, combined into an OSD video control code, and inserted into the RAM caption buffer. A preamble address code incorporates row selection in its first byte, and

attribute and indentation information in its second byte. This code causes the processor to select a new row in the RAM buffer, modify the character cursor, and set up the default video



User Console Program, DOS.  
Shown: Edit Window at left  
Teach Window at right,  
Status Window at bottom.

Manipulator Breadboard.  
Shown: 2 fingers,  
each with 2 motors.



# ROBIX™

## RCS-6 Robotic Construction System

The Robix™ RCS-6 is a 6-servo, table-top, Robotic Construction System (RCS) which can be used to build and operate robotic models. The combination of ease of construction, advanced user interface, and low cost deliver previously unavailable levels of value and performance to the robotic enthusiast. The control algorithms, user interface, and both low and high-level programming interfaces are superior to any others in this price range.

Use Robix language with built-in editor and teach-pendant mode for sequencing. Extend capability with calls from C or QB 4.5.

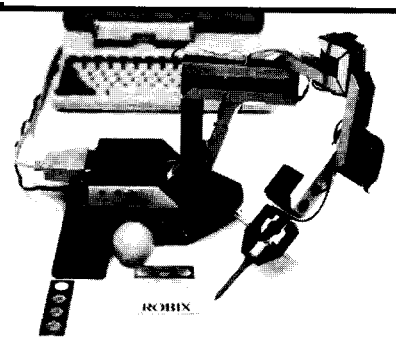
Connect to DOS PC's LPT1/2. 8-ch by 8-bit A/D, 7 switch inputs, 2 outputs. Software provides auto-matched velocity trapezoids, per-axis, maxspd, accel, decel, min/max/init/zero pos, abs/rel moves, and more.

Robix RCS-6 complete (less printer cable):

# \$550

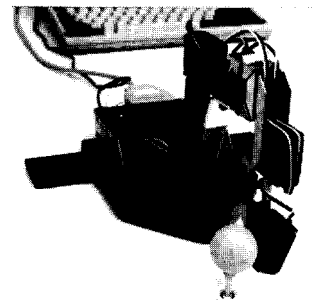
Advanced Design, Inc.

1101 E. Rudasill Road, Tucson, AZ 85718  
voice: (602) 544-2390 fax: (602) 5750703



5 Axis + Parallel Jaw Gripper.  
'Assembly time of approx 15 min,  
is typical for most constructs.

Table-Tennis Golfer.  
3 coordinated "swing" axes,  
4th axis controls club "loft".



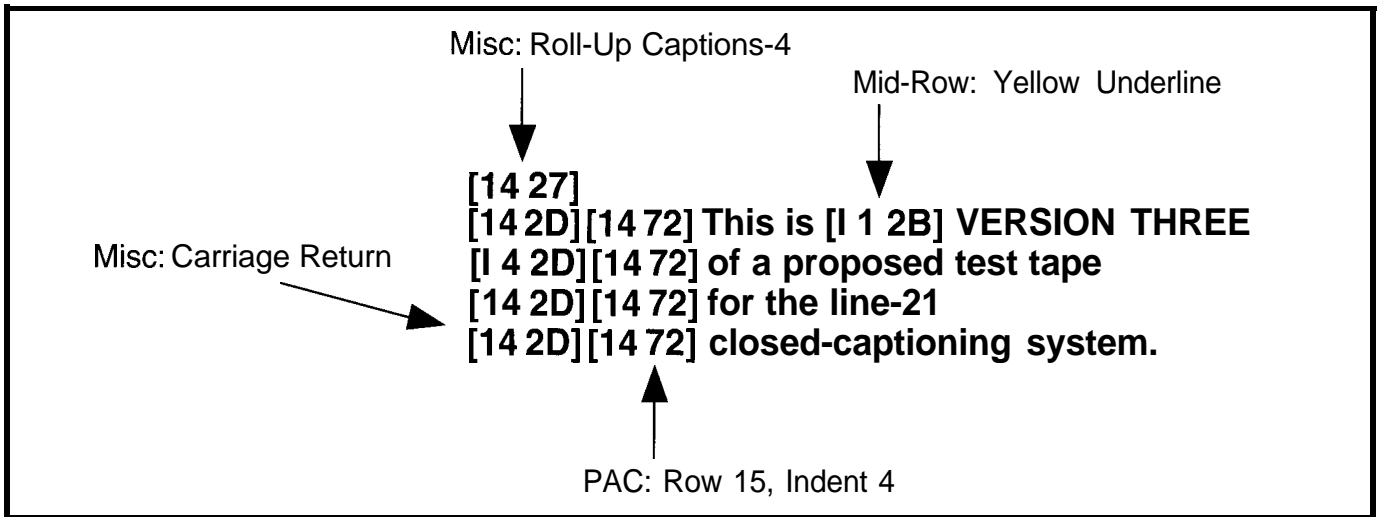


Figure 5—A sample caption from the test tape exercises several features of the closed captioning system. The final output generated by this data stream is in Photo 2.

control codes for the new row. Miscellaneous codes can be distinguished by their second bytes. They alter the RAM buffer or character cursor and must be processed on an individual basis.

The code segment featured in Listing 1 is responsible for interpreting midrow codes. M R P ROC checks for the selected display mode before calling GETATTR to extract the attributes. GETATTR converts the color, italics and underline selections into a video control code, which is passed to PUTVC (not shown) for storage in the RAM display buffer.

The OSD interrupts the data interpretation process up to four (caption mode) or eight (text mode) times per field to obtain a new row of characters to display. Once the software has determined that there is a next row to display and which RAM buffer row that it corresponds to, the routine in Listing 2 will be accessed. CP DATA copies the RAM display buffer and the video control code, matrix register, and event line to the OSD register block. Optional first and last space characters are added to the row in accordance with FCC guidelines. We balanced the needs for quick response time to the OSD and compact code writing this subroutine.

We used three tools to aid in code development. The MMDS05 development system is a compact emulator and bus state analyzer which communicates with both the target system and an IBM-compatible host computer.

It consists of a generic HC05-compatible platform that supports microcontroller-specific personality boards—in this case the M68HC05CC0 EVS. Although it lacks the bus analysis capability, the EVS was also useful as a stand-alone debugging tool. The smallest tool used in code development was the 68HC505CC1, a “piggy-back” emulator of the 68HC05CC1.

This special version of the CC1’s 42-pin SDIP package holds a CC1 internally, and accommodates the plug-in of a standard 28-pin DIP EPROM that acts as the user ROM. Use of these tools significantly decreased development time.

Putting the project together was not too difficult, with the exception of watching holes being drilled in the

**Free Demo Disk! Call 1-800-221-6630**

# Mat is C thru ROM?

ROM Your Borland or Microsoft C/C++ Code.

**C thru ROM** is the complete ROM development software tool kit. It lets you run Microsoft and Borland C and C++ programs on an embedded 80x86 CPU without using DOS or a BIOS. **C thru ROM** saves you money. There are no DOS or BIOS royalties to Pay for your embedded systems. **C thru ROM** is complete! It includes the following and much more:

- \*Supports Borland's Turbo Debugger.
- \*Remote Code View style source level debugger.
- ROMable startup code brings CPU up from cold boot.
- ROWable library in source code.
- \*Flexible 80x88 Locator.

**COMPLETE PACKAGE ONLY \$495. 30-DAY MONEY BACK GUARANTEE.**

**Datalight®**

07 N. OLYMPIC, SUITE 201 • ARLINGTON, WA 98223 • (206) 435-8086 • FAX: (206) 435-0253



cabinet of our brand new 27" color TV. Safety and space constraints forced us to mount the CC1 on a small breadboard which we attached to the television cabinet. The schematic of the CC 1 breadboard is included in Figure 4.

There were seven signals routed in short coaxial cables between the chassis and the CC1: composite video; chassis hsync and vsync; and red, green, blue, and fast blanking. The composite video was taken from the chassis video processor chip, and the other signals were connected to the existing OSD microcontroller socket. The original microcontroller was left in place to perform the general TV control functions, but the color and fast blanking signals were disconnected to prevent contention with the CC1. This meant that our modified television would no longer have regular (noncaptioned) on-screen display. In systems where the CC1 is given full control of the TV, this would not be the case.

As you can see in Photo 2, the CC1 performed with flying colors. This photo shows a caption display generated by the CC 1 from data it extracted from a test tape. The closed captioning transmitted with most shows and movies marginally exercises FCC requirements, so we relied on a set of videotapes designed to test

decoders exhaustively. The source for these tapes is listed at the end of this article. Figure 5 contains an example of the data stream on the test tape that was used to generate the caption seen in Photo 2.

## CONCLUSION

The Television Decoder Circuitry Act has opened new channels for transferring information to the viewing public. The medium for which this act was initially intended is closed captioning. Already, though, there is discussion by the TV Data Systems Subcommittee of the EIA for extended data services, which would allow for other types of information to be relayed to viewing audiences via printed text on a television screen. Examples include information about a current program such as length of feature, program title, and rating, or emergency messages from the National Weather Service.

We've given you some insights into what's required to build a closed-caption decoder, and we showed an example of a working system that uses the 68HC05CC1 to implement closed captioning. Television broadcasters will be offering more and more captioned programs in the near future, and will eventually incorporate extended data services. The CC 1, which relies on software to perform

data interpretation, has the flexibility to support both closed captioning and extended data services. □

*Janice Benzel holds BSEE and MSEE degrees from Purdue University. Linda Reuter Nuckolls holds a BSEE degree from Texas Tech University and an MSEE from the University of Texas. They both design customer-specific microcontrollers at Motorola in Austin, Texas.*

## SOURCE

Motorola Semiconductor Products  
2100 East Elliot  
Tempe, AZ 85284  
(602) 244-6900  
TWX: 910-951-1334

## REFERENCES

Television Decoder Circuitry Act of 1990, Pub. L. 101-431.

"Television Captioning for the Deaf: Signal and Display Specifications," Engineering Report No. E-7709-C, Public Broadcasting Service, 1980.

"Telection II Decoder Module Performance Specification," National Captioning Institute, Inc. 1985.

Report and Order, Federal Communications Commission GEN Docket No. 91-1, 1991.

Memorandum, Opinion, and Order, Federal Communications Commission GEN Docket No. 91-1, 1992.

"Line 21 Data Services for NTSC," Electronic Industries Association EIA-608, 1992.

### Caption Test Tapes:

The Caption Center/MARDO  
WGBH Educational Foundation  
125 Western Ave.  
Boston, MA 02134

## IRS

401 Very Useful  
402 Moderately Useful  
403 Not Useful

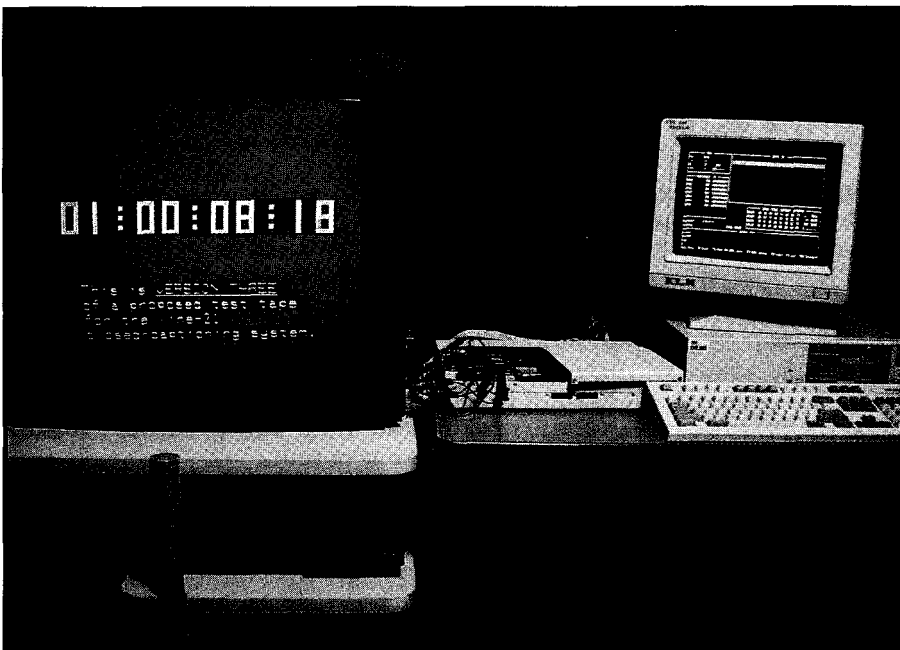


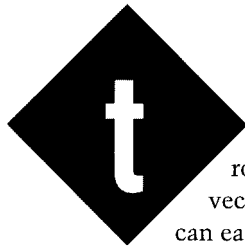
Photo 2—When installed in a new 27" television set, the **sample 68HC05CC1** circuit **performs** with flying **colors**

# Vector Approach Simplifies Three-Dimensional Graphics

Until we perfect a holographic display device, we'll have to be content projecting three-dimensional images onto two-dimensional space. Some basic knowledge of such transforms can simplify the task.

## FEATURE ARTICLE

Fred H. Carlin



Techniques routinely used in vector mathematics can ease the construction of three-dimensional drawings on a two-dimensional display device.

The calculations can be greatly simplified, and the use of trigonometric functions almost eliminated, when mapping points from an abstract, three-dimensional object space onto a two-dimensional plotting surface.

There are three basic steps you must take when creating a three-dimensional view on a two-dimensional display device. First, you need to establish a viewing plane (or screen) perpendicular to the viewer's direction of sight. Second, you need to project the points and lines from any objects in a three-dimensional object space onto the plane of the display screen. Third, you need to adjust the lengths of any projected lines or surfaces in the display plane to provide the illusion of perspective, or depth cues, where they are desired.

Figure 1 shows two independent coordinate systems that intersect one another at the origin. The first system is the object coordinate system whose axes are marked X, Y, and Z. The other is the viewer's coordinate system whose axes are marked  $X_e$ ,  $Y_e$ , and  $Z_e$ . Note that the viewer's line of sight is parallel

to the  $Z_e$  axis. The coordinates of the points along the  $X_e$  and  $Y_e$  axes form a plane perpendicular to the viewer's line of sight. This will be the plane onto which the image is projected. Many times this plane is the same plane as, or is parallel to, the screen's display surface.

Assuming the viewer's center of vision is directed at the origin reinforces the idea that the viewer's path of vision is directly along the  $Z_e$  axis. Also, the origins of the viewing coordinate system and the object coordinate system are coincident, since they intersect at the point (O,O,O). Objects not centered about (O,O,O) may be easily moved by linear translation.

In addition, we will assume the viewer wishes to see the object in the upright position. That is, the Z coordinate points toward the top of the object. [If another direction for "up" is required, then you need to specify a vector pointing in the desired direction instead of Z.]

The Z axis of the object coordinate system lies in the plane formed by  $Y_e$  and  $Z_e$ . It is necessary to locate the coordinates of this point in the  $X_e$ - $Y_e$  plane, and a mathematical property that can be used for this is the vector cross-product. The vector cross-product takes two vectors as inputs to the calculation, and yields a third vector that is perpendicular to both the

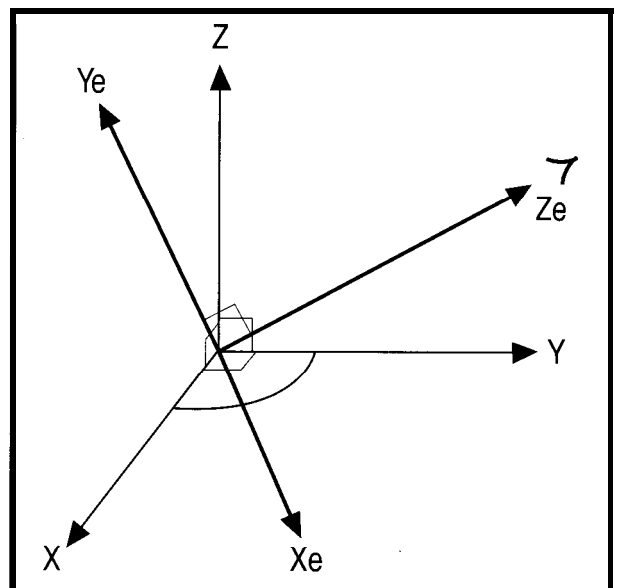


Figure 1—In any three-dimensional viewing system, there are separate object and viewer coordinate systems. In the simplest case, they both have a common origin.

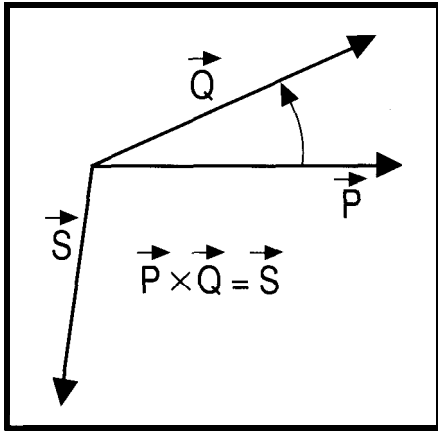


Figure 2—The vector cross-product of  $P$  and  $Q$  generates the vector  $S$ , which is perpendicular to the first two.

first and second vector. Figure 2 shows that the cross-product of vectors  $P$  and  $Q$  yields the vector  $S$  that is perpendicular to both  $P$  and  $Q$  (and hence the plane formed by  $P$  and  $Q$ ). We can obtain the vector  $X_e$  by taking the vector cross-product of vectors  $Z_e$  and  $Z$ . The vector  $X_e$  is perpendicular to both  $Z_e$  and  $Z$ . Similarly, the vector  $Y_e$  is obtained by taking the vector cross-product of  $X_e$  and  $Z_e$ . These statements can be summarized with the following mathematical expressions:

$$X_c = Z_c \times Z$$

$$Y_c = Z_c \times X_c$$

The information required to project a point from the three-dimensional object space ( $X, Y, Z$ ) to the two-dimensional plane of the display screen ( $X_e, Y_e$ ) is obtained by converting the vectors  $X_e$  and  $Y_e$  to unit vectors. A unit vector is a vector that is precisely one unit in length in a given direction. Unit vectors are obtained by dividing each coordinate of the vector by the magnitude of the vector. The equations used to calculate the unit vectors from vectors  $X_e$  and  $Y_e$  are:

$$U_x = \frac{X_c}{\|X_c\|}$$

$$U_y = \frac{Y_c}{\|Y_c\|}$$

I will illustrate the application of these formulae in the following example. Suppose the viewing position of an object lies along the vector  $Z_e$

and is at the point  $x=-3, y=4,$  and  $z=5$  in the object coordinate system. Then  $X_e$ , a vector perpendicular to both  $Z$  and  $Z_e$  (which is a vector from the viewing position to the origin) is found as follows:

$$X_c = Z_c \times Z$$

$$X_c = \begin{vmatrix} i & j & k \\ -3 & 4 & 5 \\ 0 & 0 & 1 \end{vmatrix} \\ = (4-0)i + (0+3)j + (0-0)k \\ = 4i + 3j + 0k$$

The next step is to compute the vector  $Y_e$  using the just derived vector  $X_e$  by using it in the cross-product formula with vector  $Z_e$ . These calculations are shown below:

$$Y_c = Z_c \times X_c$$

$$Y_c = \begin{vmatrix} i & j & k \\ -3 & 4 & 5 \\ 4 & 3 & 0 \end{vmatrix} \\ = (0-15)i - (0-20)j + (-9-16)k \\ = -15i + 20j - 25k$$

The unit vectors  $U_x$  and  $U_y$ , which are derived from the vectors  $X_e$  and  $Y_e$  respectively, are calculated as follows by applying the unit vector formula to each vector independently:

$$U_x = \frac{4i + 3j + 0k}{\sqrt{16+9+0}} \\ = 0.8i + 0.6j + 0k$$

$$U_y = \frac{-15i + 20j - 25k}{\sqrt{225+400+625}} \\ = 0.424i + 0.566j - 0.707k$$

With these points calculated we are ready to go on to the calculation of the points to be plotted on the screen.

## AXOMETRIC DISPLAY

Projecting a point  $P(x, y, z)$  onto the viewing screen requires calculating the plot screen coordinate values  $X_{plot}$  and  $Y_{plot}$ , which are the projected equivalent values for the point  $\{U_x, Y_x\}$ . These are calculated by using the dot product of each of the unit vectors with the point  $P$ :

$$X_{plot} = P \cdot U_x$$

$$Y_{plot} = P \cdot U_y$$

Figure 3 shows two vectors  $P$  and  $Q$  with their included angle, designated as  $\theta$ . The dot product formula shown above for these vectors is also equivalent to the following formula:

$$P \cdot Q = \|P\| \|Q\| \cos \theta$$

If  $Q$  is a unit vector (defined as having a length of exactly 1 unit) and lies on the principal axis of the object coordinate system,  $Z$ , then application of the dot product will yield the distance that the point is along the  $Z$  axis, or away from the  $X$ - $Y$  plane. This distance is labeled as " $S$ " in Figure 3. The value of  $S$  will be positive if  $P$  is on the same side of the  $X$ - $Y$  plane as  $Q$  and negative otherwise.

I will illustrate the use of the dot product to calculate the screen coordinate values,  $X_{plot}$  and  $Y_{plot}$ , by continuing with the same point coordinates as I used in the previous example, the plot values for a point  $(1, 2, 3)$  (or vector end point  $\{i + 2j + 3k\}$ ) is:

$$X_{plot} = P \cdot U_x \\ = (i + 2j + 3k) \cdot (0.8i + 0.6j + 0k) \\ = 0.8 + 1.2 + 0 \\ = 2.0$$

$$Y_{plot} = P \cdot U_y \\ = (i + 2j + 3k) \cdot (-0.424i + 0.566j - 0.707k) \\ = -0.424 + 1.132 - 2.121 \\ = -1.413$$

As a result, the point at  $(1, 2, 3)$  would project onto the screen at the  $X$ - $Y$  coordinate point  $(2.0, -1.413)$ . All you have to do now is to scale or adjust the plot value for the screen or plotter coordinates.

Recall, the unit vector calculations need be performed only once for any viewing position in three-dimensional space. The  $Z_{plot}$  is ignored in the axometric display, which results in collapsing the  $Z$  dimension.

## PERSPECTIVE DISPLAY

A perspective display is generated in a similar manner to the axometric display. In the perspective display system, the distance from the observer is taken into account and the object scaled appropriately. This is easily accomplished with some additional

# IMAGENATION...

## VIDEO FRAME GRABBERS

- ◆ Simplicity
- ◆ Functionality
- ◆ Affordability
- ◆ Accuracy

### Cortex-I

**\$495<sup>00</sup>**

- ◆ Real-Time Capture
- ◆ Half Slot XT/AT
- ◆ 512 x 484 x 8 Bit
- ◆ RS-170/CCIR
- ◆ External Trigger

### Cortex-STD

**\$595<sup>00</sup>**

- ◆ Dual Video Input
- ◆ Opt. XMS Mapped
- ◆ Low Power Options
- ◆ STD-80 or 32 Bus
- ◆ External Trigger

### Vidmux-4

**\$99<sup>00</sup>**

- ◆ 4 to 1 MUX
- ◆ Half Slot XT/AT

### Includes Software...

- ◆ C Library & Source
- ◆ Image Capture Utility
- ◆ Tiff Utilities
- ◆ "Image" Drive Ram Disk Emulation

### STARTER PAK...

#### Cortex-Pak

**\$1395<sup>00</sup>**

- ◆ NEC/TI-23EX Camera With Lens
- ◆ 9" Video Monitor
- ◆ Frame Grabber
- ◆ Software & Cables

OEM PRICING AVAILABLE

## IMAGENATION CORP.

P.O. Box 84568  
Vancouver, WA 98684

PH/FX (206) 944-9 13 1

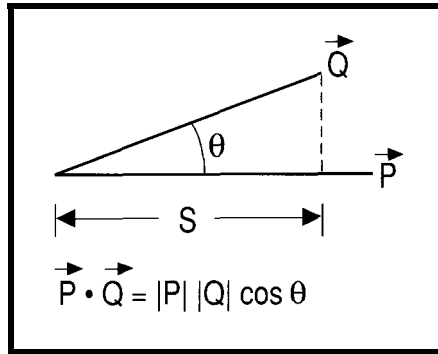


Figure 3-The vector dot product or scalar product projects the shadow of one vector on to the other.

information regarding how far the point is from the viewing screen. This is done by first finding the unit vector perpendicular to the  $X_e$ - $Y_e$  plane or screen. This vector has been previously established and its unit value is simply:

$$U_z = \frac{-Z_c}{\|Z_c\|}$$

The  $X_{plot}$ ,  $Y_{plot}$ , and  $Z_{plot}$  are formed by the dot product of the vector described by the point  $P$  and the unit vectors of the viewing coordinate system. Perspective is added by adjusting the  $X_{plot}$  and  $Y_{plot}$  values by the distance that particular point is from the observer.

The reasoning behind this is shown in Figure 4. It shows a diagram of an observer located a distance  $D$  in front of the viewing plane. The viewing plane passes through the origin and is shown in the figure edge-on. A typical point  $X_{plot}$  is shown transformed and plotted in three dimensional space (i.e.,  $Z_{plot}$  units behind the viewing screen).

Note that to keep Figure 4 simple, the  $Y$  axis values are not shown in the figure. This being the case, all

calculations for any of the  $Y$  values are also omitted from the discussion. The arguments and computational methods for the  $Y$  values are identical to those for the  $X$  values except the variable values for any  $Y$  value are substituted for each  $X$  value.

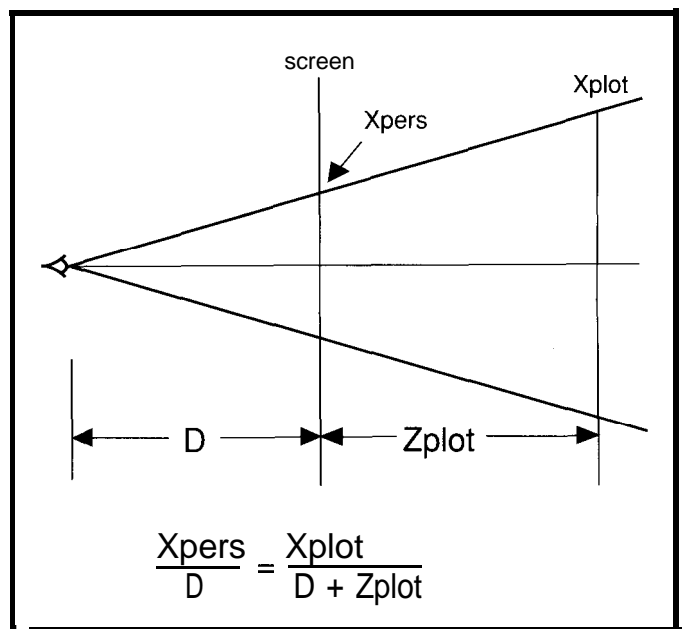
Perspective scaling is obtained by using the following relation:

$$\frac{X_{pers}}{D} = \frac{X_{plot}}{D + Z_{plot}}$$

$D$  is the distance the observer is from the origin of the viewing plane.  $Z_{plot}$  is the distance from the viewing plane to the point to be displayed. I can, referring to our example, calculate the unit  $Z$  vector or  $U_z$  by using the following:

$$\begin{aligned} U_z &= \frac{-Z_c}{\|Z_c\|} \\ &= \frac{-(-3i + 4j + 5k)}{\sqrt{9 + 16 + 25}} \\ &= \frac{+3i - 4j - 5k}{7.07} \\ &= 0.424i - 0.566j - 0.707k \end{aligned}$$

The  $X_{pers}$  and  $Y_{pers}$  values (i.e., the ones used to plot points on the screen) are found by first calculating the distance the viewer is from the screen. This distance is used in the relations that are used to adjust the values of  $X_{plot}$  and  $Y_{plot}$ . These operations are shown below:



$$\frac{X_{pers}}{D} = \frac{X_{plot}}{D + Z_{plot}}$$

Figure 4-Scaling for perspective is proportional to distance from the viewing screen or plane.

## VECTOR MATHEMATICS

A three-dimensional vector can be represented as an array of three numbers, each corresponding to one axis of the three-dimensional coordinate system. For example:

$V = (1, 2, 3)$  is the notation for a vector from the origin to the point  $X = 1, Y = 2, Z = 3$

This vector can also be written in its equation form as:

$$V = 1i + 2j + 3k$$

When vectors are represented in this form, then the variables  $i, j,$  and  $k$  are tags to label the values corresponding to the respective values for  $X, Y,$  and,  $Z.$

## ADDITION

The addition of two vectors is accomplished by adding their corresponding parts. For example:

$$\begin{aligned} U &= 1i + 2j + 3k \\ V &= 4i + 5j + 6k \\ U + V &= 5i + 7j + 9k \end{aligned}$$

## MULTIPLICATION

There are two types of vector multiplication: The dot product and the cross product. Each has special properties that can be used in three-dimensional plotting.

## DOT PRODUCT

The result of a dot product is a simple numeric value or scalar. The dot product is numerically equal to the length of vector  $A$  times the length of vector  $B$  times the cosine of the included angle.

Graphically, the dot product projects the first vector onto the second and the numeric value is the length of the projection or "shadow." For example:

$$\begin{aligned} A &= (a_1)i + (a_2)j + (a_3)k \\ B &= (b_1)i + (b_2)j + (b_3)k \end{aligned}$$

Then the dot product is easily computed from:

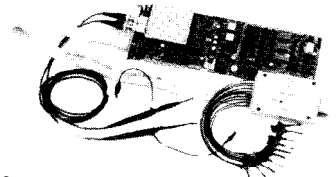
$$A \cdot B = (a_1)(b_1) + (a_2)(b_2) + (a_3)(b_3)$$

## CROSS PRODUCT

The result of a cross-product is a new vector. This vector has special properties that make it very useful in three-dimensional graphing applications: First, its length is equal to the length of vector  $A$  times the length of vector  $B$  times the sine of the included angle. Secondly, the new vector has a direction perpendicular to both vectors  $A$  and  $B.$  An example of a cross-product calculation is shown below:

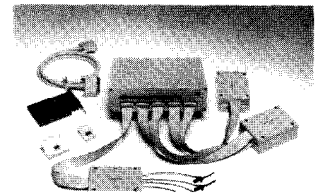
$$\begin{aligned} A \times B &= \begin{vmatrix} i & j & k \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} \\ &= [(a_2 b_3) - (a_3 b_2)]i - [(a_1 b_3) - (a_3 b_1)]j + [(a_1 b_2) - (a_2 b_1)]k \end{aligned}$$

## 200MSa/S Digital Oscilloscope



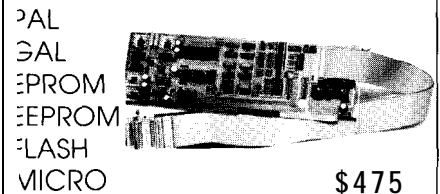
- 200 MSa/S sampling rate
  - PC-BASED INSTRUMENT
  - 2 Analog channels
  - 8 Digital channels (8ch. logic analyzer)
  - 125MHz Single shot Bandwidth
  - 4K samples/channel (analog & digital)
- \$1599 - DSO-28 100 Price is Complete  
\$1999 - DSO-28200 Pods and Software included.

## 400 MHz Logic Analyzer



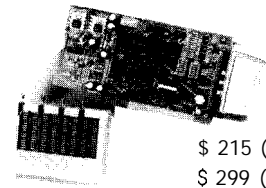
- up to 128 channels
  - up to 400 MHz
  - 16K samples/channel
  - Variable threshold
  - 8 External clocks
  - 16 level triggering
- \$799 - IA12100 (100 MHz, 24 Ch) Price is Complete  
\$1299 - LA32200 (200 MHz, 32 Ch) Pods and Software  
\$1899 - LA32400 (400 MHz, 32 Ch) included

## Universal Programmer



- PAL  
GAL  
EPROM  
EEPROM  
FLASH  
MICRO
- \$475
- 5ns PALs  
4 MEG EPROM (8 & 16 bit)  
22V 10 & 26CV12 GALs  
Free software updates on BBS

## Gang Programmer



- \$ 215 (4 Sockets)  
\$ 299 (8 Sockets)

Up to 1 MEG EPROMS

Call (201) 808-8990  
Link Computer Graphics, Inc.  
369 Passaic Ave, Suite 1100, Fairfield, NJ 07004 fax. 808-8786



$$D = || Z_c ||$$

$$= \sqrt{9 + 16 + 25}$$

$$= 7.07$$

$$Z_{plot} = P \cdot U_z$$

$$= (i+2j+3k) \cdot (0.424i-0.566j-0.707k)$$

$$= -2.829$$

$$X_{pers} = \frac{X_{plot} \times D}{D + Z_{plot}}$$

$$= \frac{2.0 \times 7.07}{7.07 - 2.829}$$

$$= 3.334$$

$$Y_{pers} = \frac{Y_{plot} \times D}{D + Z_{plot}}$$

$$= \frac{-1.413 \times 7.07}{7.07 - 2.829}$$

$$= -2.356$$

Note that the  $X_{pers}$  and  $Y_{pers}$  values are larger than the corresponding  $X_{plot}$  and  $Y_{plot}$  values because the point is closer to the viewer than the viewing screen formed by  $X_e$  and  $Y_e$ .

## CONCLUSIONS

The vector and point manipulation equations that I used in this discussion often form part of the core of complex rendering and image generation applications that run on a variety of computing platforms. Even though these techniques are often introduced in a second-semester calculus course, the methods and concepts that are used in these equations should be understandable to anyone who has a full grasp of trigonometric concepts and operations. While these equations look like they wouldn't be too compute intensive, when you run them over the many thousands of points contained in a typical high-resolution display, and then consider color, motion, and complex solids that may have many surface attributes, you can begin to appreciate the reason why high-powered programs like the one used to create the special effects in the motion picture Terminator II often run on fast, expensive, and very powerful machines. ☐

*Fred Carlin is a consulting engineer in Goleta, Calif. In addition to designing hardware and software for real-time data-acquisition and -processing systems, he has developed systems for medical diagnostics, environmental monitoring and industrial control. Fred holds Ph.D. and M.S. degrees from the University of California.*

## REFERENCES

References to vector mathematics can be found in most freshman college-level mathematics texts on analytical geometry. Some typical texts are:

George B. Thomas: *Calculus and Analytical Geometry* (Reading, MA: Addison-Wesley, 1969).

A. W. Goodman: *Analytical Geometry and the Calculus* (New York: Macmillan Co., 1965).

## IRS

404 Very Useful  
405 Moderately Useful  
406 Not Useful

# 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: (919) 667-2100  
Fax: (919) 667-0487

# Graphics LCD Control for Embedded Applications

In most controller applications, precious little free processor time is left to devote to display functions. Off-load the bulk of the display tasks to a custom controller chip and see what magic can be done.

## FEATURE ARTICLE

David Erickson

**O**his article describes my hardware/software implementation for a Graphics LCD display on an embedded controller. The approach I took includes the design of a Programmable Gate Array (PGA) that I use to control an LCD. I implemented the software in Small C and 68HC11 assembly. It seems to me that just about any application can be enhanced with a graphical interface. Many products already use PC graphics or TV video as their display. But a lot of these products are big and tend to stay put in one place.

I wanted to extend the range of products that display with graphics to include low-cost, low-power, and portable applications. For several years I have been intrigued by the possibility of using a graphics LCD display for

such applications. I enjoy building electronics relating to my sailing and wind surfing passions. After having built a few small hardware-only projects, I was frustrated by having to repackage the project each time I wanted to add even simple increments in functionality. Enter microprocessors. Using the power of microprocessors, I built an MC6803-based weather station with a two-line LCD readout and LEDs arranged in a circle to display the wind direction. But before long, I was frustrated by the inability to display trends and by the need to code complex functions in assembler. I was doing some pretty heavy stuff in assembler-for a hardware guy. It was when I began using lots of pointer math on a CPU with only one pointer register that my coding productivity declined-fast. I gave up on doing anything as ambitious as graphics programming until I could do it in C. But buying a C compiler for a micro was well beyond my limited budget.

For a while I was tempted to build systems around PC components just to use the great development tools. But PCs and boats don't get along well: Disk and keyboard versus salt water, inadequate power, no easy way to remote an LCD display, cost.. .the list goes on.

Then three things happened. Ads for surplus graphic LCDs began

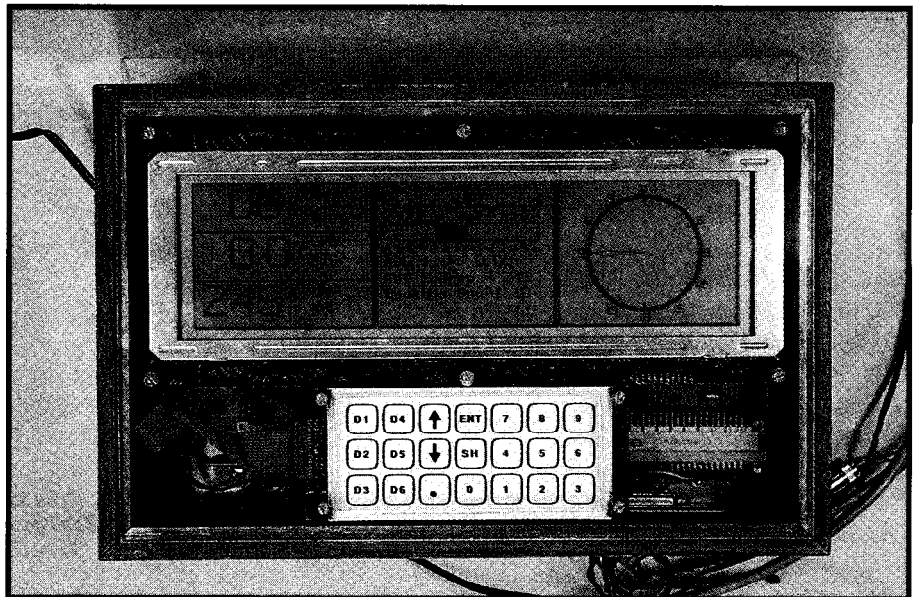


Photo 1-The LCD1 custom controller chip makes a sophisticated display possible while using little main processor overhead. One application for the LCD controller chip is in a boat navigation system.

Address	Reg Name	Bits	R/W	Range	Description
0,1	HPOS	9	W	0-51 1 (0-479 displayed)	Horizontal Pos.
3	VPOS	8	W	0-255 (0-1 27 displayed)	Vertical Pos.
4	DATA	8	R/W	MSB=Right, LSB=LEFT	Byte Data R/W
5	OPERATION	3	W	000—Pixel ON 001—Pixel OFF 01 0—Pixel XOR 01 1—Unused 100—Byte OR 101 —Unused 11 0—Byte XOR 11 1—Byte Write	Drawing Operation Mode
6,7					Unused

Table 1—The LCD's functions are accessed through a series of control registers. Note that address 2 is available for future LCD panels with more than 256 pixels vertically. Addresses 6 and 7 are a/s/o unused.

appearing. Motorola made the Small C compiler for the 68HC11 available, and *Circuit Cellar INK* published an informative and useful article on the Motorola 68HC11 (issue #24, Dec. '91/

Jan. '92). I immediately began to collect the pieces to do my dream project: "BoatBus," a digital display system for my sailboat. Getting hardware and software tools going for

almost "no money down" was easy, thanks to Motorola. In no time I was writing in C, using a real debugger (the Buffalo monitor), and using a modern, powerful \$12.00 micro. I even got the floating-point math library (MATH 11) working!

I purchased an LCD display from Timeline, an Hitachi LM215XB with 480x128 pixels and a separate controller board. Hooking the board up and getting basic graphics working happened in a few evenings.

I was on a roll! Starting with `putPix` and `putByte`, I soon had line draw, polyline, and characters being displayed. Then came large font characters and display of bit-mapped pictures (created on a real computer).

The application was also doing well. After about six months of one or two evenings a week, I had a digital

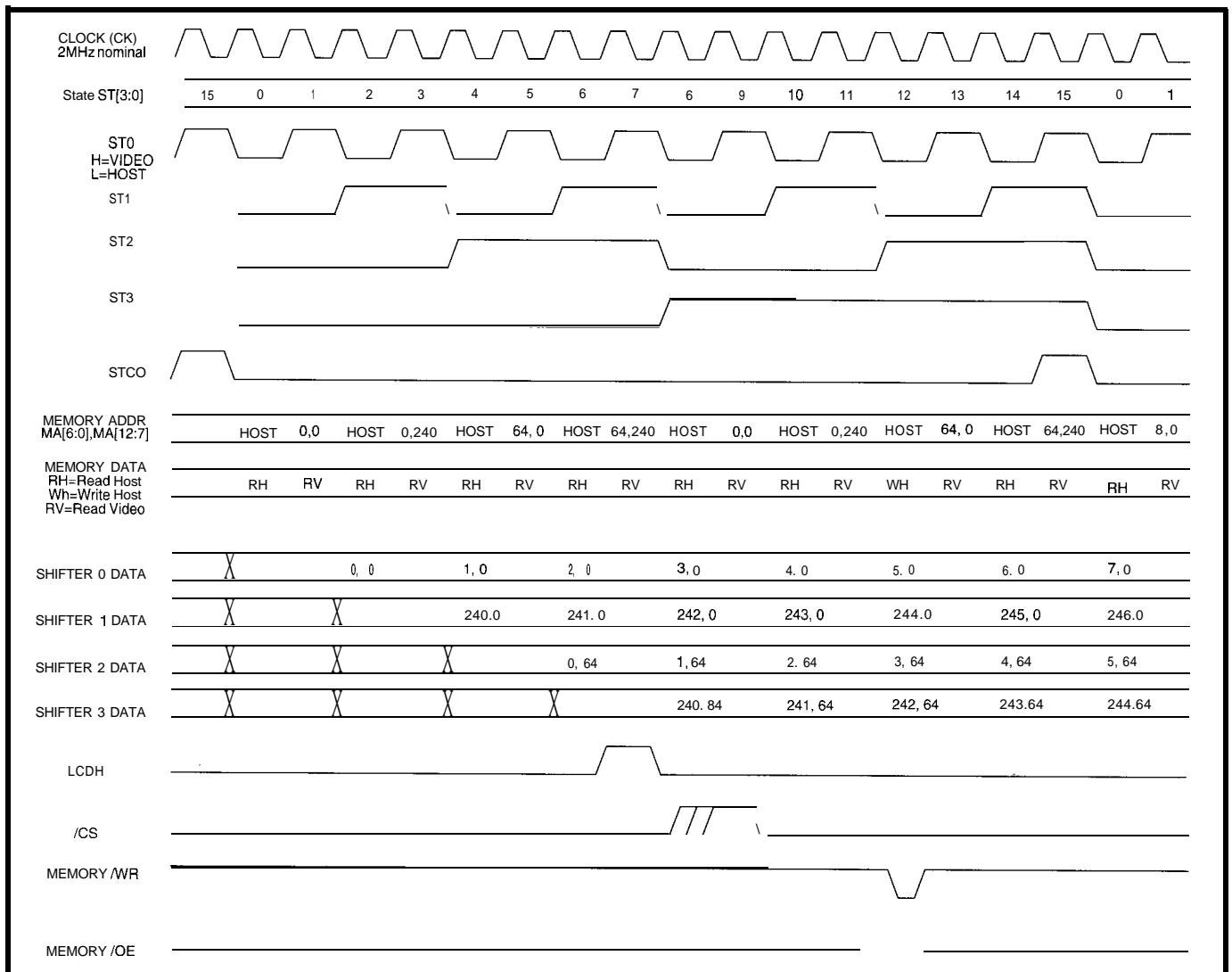


Figure 1—Memory and LCD data path timings

compass, display of wind speed and direction, boat speed, distance, battery voltage, and a few other bells and whistles. BoatBus's maiden voyage was a two-week trip to Maine during which it helped us to navigate in the fog (to the island where we sat out hurricane Bob, but that's another story).

## LCD CONTROLLER WOES

Having a winter between sailing seasons to dream up ways to improve BoatBus, I was nagged by deficiencies in the LCD controller. Although adequate for now, there were several faults. The only available controller for the LM215XB display uses two Hitachi 61830 chips plus four SRAMs. This is available neatly packaged on a PC board which has a standard eight-bit interface that ties in easily to any micro.

There are problems with this arrangement, though. Each chip controls one half (left or right) of the panel. This makes graphics operations painfully slow. For instance, to write a single pixel, first figure out which chip to access by seeing if *H* is greater than 239. Then calculate the memory address using the formula  $[(V \times 30) + (H / 8)]$  and send it to the chip. Then send the three least-significant bits of *H* to the pixel to set or clear the register depending upon whether you want to turn the pixel on or off. Each write to the chip requires you to first write to the control register and then to the data register (similar to an alphanumeric LCD). The grand total for this single pixel is 29 assembly instructions to write a byte and 37 to write a pixel, including an eight-bit multiply.

As I explored the LCD panel and controller market, I discovered that each panel manufacturer has either its own, or recommended a unique, controller. The LCD control signals are similar but different between manufacturers and even between generations from a single manufacturer. Every controller has a different host interface and uses different software. Some controllers use direct memory mapping and therefore tie up significant memory space on a 16-bit

address bus. There is almost no consistency between chips or LCD panels. It became clear that as a user of surplus LCDs, I would be switching software, hardware, and packaging often. This prospect did not seem to be a pleasant one.

## ENTER LCD1

As an avid Programmable Gate Array (PGA) user at work, I considered designing an LCD controller chip. In the past, I have designed Xilinx, Altera, and Actel chips for several applications. This was going to be the

first PGA design for my own use, though.

I call this chip design LCD1. The goals of this design are to arrive at a low-cost, single-chip (plus RAM), low-power LCD controller with an efficient hardware and software interface for low-cost microprocessors.

As I was both the hardware and the software engineer on this project, and having used an unsatisfactory controller, I knew exactly what I wanted (and didn't want) for the LCD controller. First, there should be no software address mapping. The

## An Unbiased Survey for DOS Developers.



# No!



I don't want to find out how I can save a lot of money using ROM-DOS 5 instead of **MS-DOS®** in our 80x86 product line. I don't care if ROM-DOS 5 is incompatible with MS-DOS 5 but costs much less. I like spending much more than I have to. It makes me feel like a philanthropist and besides Microsoft@ probably needs the money more than I do anyway.



Yes, I want to know the facts about ROM-DOS 5. Please send me information and a free bootable demo disk to try with my software.

**In the U. S. A. Call Toll Free 1-800-221-6630**  
**or fax this coupon to (206) 435-0253.**

Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

city \_\_\_\_\_ State \_\_\_\_\_ zip \_\_\_\_\_

Phone \_\_\_\_\_ Fax \_\_\_\_\_

307 N. OLYMPIC, SUITE 201 • ARLINGTON, WA 98223 USA • (206) 435-8086 • FAX: (206) 435-0253

# Datalight®

MS-DOS and Microsoft are registered trademarks of Microsoft Corporation

controller should be programmed directly in terms of *H* and *V*. Second, registers should be directly mapped to the host. Registers that are greater than eight bits should appear as two contiguous bytes so that a STAD (16-bit write) instruction can be used. Pixel drawing operations (SET, C LR, X0 R) should be done in hardware. Pixels are to be written by setting a drawing mode (SET, C L R, X0 R) and then feeding the *H* and *V* coordinates to the controller. The pixel writes are triggered by the *H* address being written. Bytes are written by setting a drawing mode and then feeding *H* and *V* coordinates and bytes of data to the controller. Data writes are triggered by the data being written. Operations should be fast enough so that no waiting for a status flag is required. Table 1 contains the register set I settled on. Address 2 is available for future LCD panels with more than 256 vertical pixels.

An additional goal is that future LCD panels should use the same hardware and software interface. I change panels so seldom that designing new timing circuitry should be all that is required to support a new LCD.

## START WITH THE MEMORY TIMING

I began the design of the basic memory cycle to support the LM215XB. Like most LCDs, the interface consists of a clock, *H* and *V* syncs, and data. The challenge with using this panel is that each of the four data wires drives one of the four quadrants on the panel. Each quadrant can be thought of as a 240x64 pixel raster for a total of 480x128 pixels. There is no horizontal or vertical blanking on most LCDs. Simply output an *H* sync for every 240 clocks, and a *V* sync for each 64 *H* syncs.

Figure 1 shows the memory and LCD data path timing. The LCD clock frequency recommended by Hitachi is roughly 1 MHz. This clock frequency allows the LCD to be refreshed every (240x64x1μs) 15.3 ms, which is equivalent to a 65-Hz sweep. Since four bits are output each microsecond, the total data rate the LCD needs is 4 megabits per second. By reading eight bits at a time from a single SRAM, this

requires an access once every two clock cycles (2 μs). One of my requirements is to have no-waiting-host performance. Since pixel operations require that memory be read, modified, and then written to the display, I settled on a memory cycle time of 500 ns with every alternate cycle used for video refresh and host access. This is twice the minimum requirement for the video port and allows a pixel write operation to happen in four cycles or a respectable 2 us. The memory access-time requirement is a lethargic 300 ns.

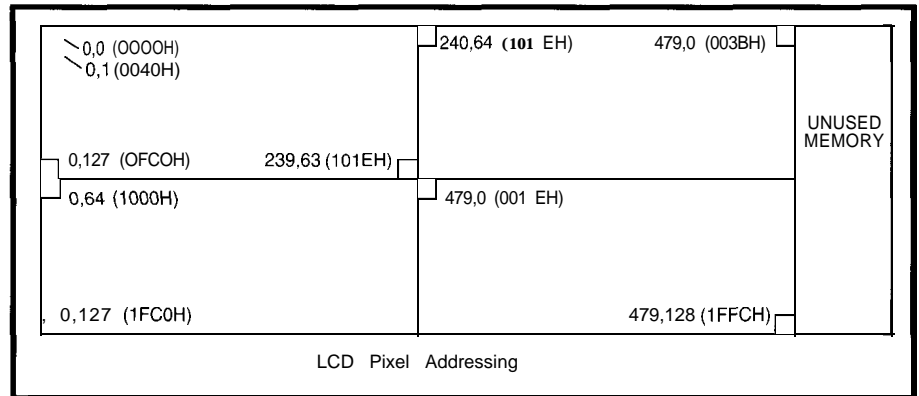


Figure 2—The LCD display is broken into four sections, so memory isn't contiguous across the entire display, making pixel address to memory address translation difficult.

Going faster is a possibility, but would cost some additional power.

## ADD THE ADDRESSING

Figure 2 shows how pixels are mapped in memory. The most difficult part about using the LM215XB display is its arrangement of four quadrants. To output four bits of data to the display, the bits must be read from four different parts of memory. This reading could be done in software by making the host addressing more complicated, but one of my goals was to simplify host addressing.

I chose to bite the bullet and do the video addressing in hardware. Since the LCD is 480 pixels across, I used 512 bits (64 bytes) per line, wasting the 32 pixels at the end of each line. By making a display line a binary multiple, the vertical and horizontal addressing can be directly wired into the address lines of the RAM. There are 64 bytes, or six address lines, for *H* and 128 bytes, or seven wires, for *V*. The total is 13 lines or 8 Kbytes total. A single 6264-type SRAM does nicely.

Figure 3 is the block diagram for LCD1. The video-address logic is implemented with two counters: 30 states for *H* and 64 states for *V*. Since the LCD requires that data be read out of four quadrants at a time, I used adders to calculate the addresses. Bytes *H*, *V*; *H*+30, *V*; *H*, *V*+64; *H*+30, *V*+64 are read from memory in four successive video memory cycles. To add 30 to the *H* count, I used a five-bit adder. To add 64 to *V*, I simply set bit six to a 1. Both the *H* and *V* counters are synchronous. *H* is five bits, *V* is six bits,

Figure 1 is a timing diagram for one set of memory cycles. After every sixteen clock cycles, or eight pixels, the *H* address is incremented. The sixteen cycles are defined by ST[3:0], which is a state counter. This is implemented as a synchronous four-bit up-counter with a carry out. The carry out enables the *H* counter to increment. All the memory and data path control signals are decoded off this counter.

## SERIALIZE THE PIXELS

Each quadrant needs serial pixels, but the data comes out of the RAM eight consecutive pixels at a time in parallel. Originally, I used four 8-bit parallel-in, serial-out shift registers. Each shift register was loaded once every eight pixel times in sequence. Because the data from all four shift registers is needed simultaneously, I used three D flip-flops to delay the output of the first register, two for the second, one for the third, and none for the fourth.

Later on in the project, in a gate reduction frenzy, I used up the spare

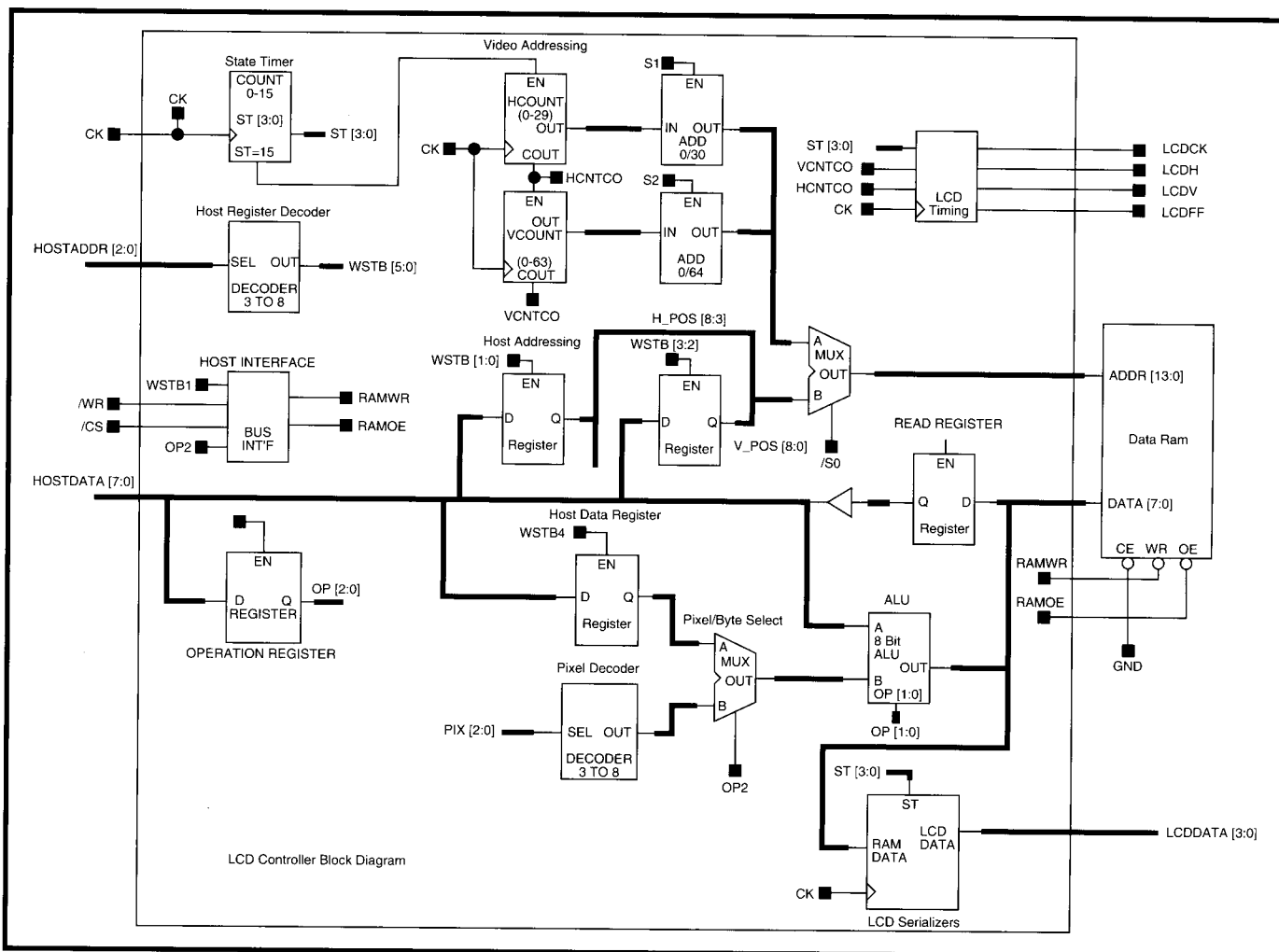


Figure 3—The LCD1 controller chip is drawn in a hierarchical fashion. At the top level, all the functional blocks of the chip can be seen at once.

memory bandwidth by loading only four bits at a time, but doing it twice. Four 4-bit shift registers fed by a 4-bit 2:1 multiplexer use a lot fewer gates than four 8-bit registers.

## ADD THE HOST DATA PATH

The drawing operations I wanted to include are: pixel set, pixel clear, pixel XOR, byte write, byte OR, and byte XOR.

To set a single pixel in a byte-wide RAM, it is necessary to read a byte from memory, OR that with a byte containing the desired pixel and write the byte out again. This would take several assembly instructions, but it is a good job for logic.

To clear a pixel requires the same read, but this time the byte containing the pixel must be inverted and ANDed before being written. Setting and clearing a pixel is the minimum set of functionality. XOR is also really useful for graphic operations, so I added it to

the drawing repertoire. The hardware for the data path is fairly simple. A single register holds the data read from the last RAM host-cycle. Byte operations use a byte from the host, written into the host write-register. Pixel operations are done by setting a bit which is selected by decoding the three least-significant bits of the horizontal address. A simple 3:8 decoder does this. For byte operations, the write-register is selected. For pixel operations, the pixel decoder is selected. This data is merged with the read data and sent back to RAM on the next memory cycle.

The ALU does simple bitwise (logical) operations. I guess that makes it an LU. Anyway, Table 2 shows the drawing operations and the logic required. Note that there are only four different ALU operations and two A-operand choices. This fit nicely in a 3-bit opcode register, OP[2:0], which is written to by the host.

The A-operand select is simply an 8-bit-wide 2:1 multiplexer. The ALU consists of eight 4-input multiplexers, each of which selects a logical combination of the A and B operands. Figure 4 is the schematic for a 1-bit channel of the 8-bit ALU.

## STIR IN SOME CONTROL LOGIC

We're almost there. Once I designed the addressing, data path, and timing, it was a lot easier to line up the LCD timing to the data than it was to move the data. I used the Horizontal-carry-out of the H-counter plus a decoded state of the state counter to generate the H-sync to the LCD. For V-sync it was necessary to decode state 0 of the Vertical counter. LCDs, being AC devices, also require a square wave at half the vertical frequency. This was simply a flip-flop that is toggled once on every vertical-sync pulse.

To perform any write operation, the host bus will always perform read,

Drawing operation	A operand	B operand	Logical Operation
Pixel set	Decode H[2:0]	Read Data	A OR B
Pixel clear	Decode H[2:0]	Read Data	NOT A AND B
Pixel XOR	Decode H[2:0]	Read Data	A XOR B
Byte WRITE	Host Write Data	Don't care	A
Byte OR	Host Write Data	Read Data	A OR B
Byte XOR	Host Write Data	Read Data	A XOR B

Table 2—Each typical drawing operation can be directly translated to a logical operation to be performed within the controller chip.

modify, and write cycles. This required a simple state machine to receive an initiation pulse, synchronize it, then read the RAM, wait for the ALU to do its thing, and write the RAM. Reads of the RAM are performed every host memory-cycle that is not a write cycle. This way, the Read data register always contains the byte pointed to by the latest writes to the HPOS and VPOS registers. The state machine controls the data transceiver and the RAM's output enable and write lines. The host write-cycle is triggered on writes of the H-position register for pixel reads. For byte reads, it is triggered on data-register writes.

## THE CHIP AND THE TOOLS

My desire to use a single, low-cost, low-power device led me to use Actel. Xilinx LCAs are RAM based and require an (E)PROM to hold their configuration data. Their advantages include that they are low power and can be reprogrammed by changing the PROM. Altera MAX family chips are great for very high speed work, but like most fuse-based logic, they are fairly power hungry.

Running at only 2 MHz, speed is a nonissue. I regularly squeeze 10 and 20 MHz out of these devices. In fact, not having to worry much about speed is a luxury that makes designing a plea-

sure. To see if the design would fit in an Actel Chip, I drew a block diagram and began adding up logic modules. My initial module count estimate was 350. The A1010 has 295 and the A1020 has 525. Unless I was way off, it would easily fit in the larger of Actel's first generation devices, the A1020. I might even be able to squeeze it into the lower-cost A1010.

Actel, like Xilinx, offers different density chips in the same package. In this case, both chips come in a 68-pin PLCC and have identical pinouts. Perfect.

Each Actel logic module is equivalent to between one and three two-input gates. It takes one module to build a latch and two to build a D flip-flop. Examples of things that can be made of one cell are: an XOR followed by a NAND, a two- or four-input multiplexer, or a D-latch. Examples of things that can be made from two cells are: a D-type flip-flop with enable, a D-flop with a two-input multiplexer on the D input, or a J-K flip-flop. A 74161 counter costs about

It's an Analog World out there

Introducing Micromint's new Temperature Sensors and Signal Conditioners

**TP1** -40 F to +230 F  
LM34-based temp. probe w/  
3-foot shielded lead

**TP2** -40 C to +110 C  
LM35-based temp. probe w/  
3-foot shielded lead

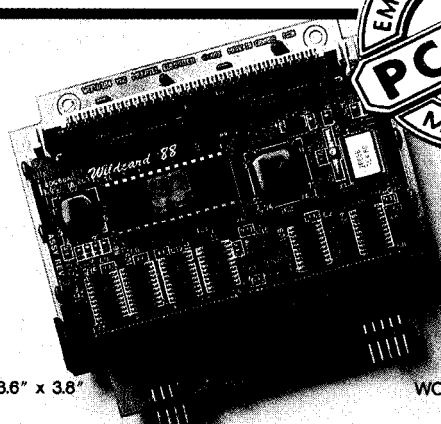
**Signal Conditioner**  
(call for specific part number)  
Input range:  $\pm 100$  mV,  $\pm 500$  mV,  
 $\pm 1$  V,  $\pm 5$  V,  $\pm 10$  V, 0-20 mA, 4-20 mA  
Outputs: -5 to +5 V, 0 to +5 V

To Order Call:  
**1-800-635-3355**

4 Park Street • Vernon, CT 06066  
(203) 871-6170 • Fax (203) 872-2204  
in Europe: 0285-658122 • in Canada: (514) 336-9426  
in Australia: (02) 888-6401 • Distributor Inquiries Invited!



# Embedded PC Module



**EMBEDDED-PC  
PC/104  
MODULES**

**\$99**  
Qty. 1

3.6" x 3.8" WCPU/104

- PC/104 Compatible 80C88® 8Mhz CPU Module
- 256K Memo& (up to 640K)
- 32K BIOS ROM
- Keyboard Port-Speaker Port

For more information please call:  
125 Wendell Ave., Weston, Ont. **(416) 245-3324**  
M9N 3K9 Fax: (416) 245-6505

Megate! is a registered trademark of Megate! Computer Corp.  
80C88 is a registered trademark of Intel Corp.

**megate!**

Figure 4—The ALU consists of eight 4-input multiplexers, each of which selects a logical combination of the A and B operands. A 1-bit channel of the ALU is duplicated eight times for the final unit.

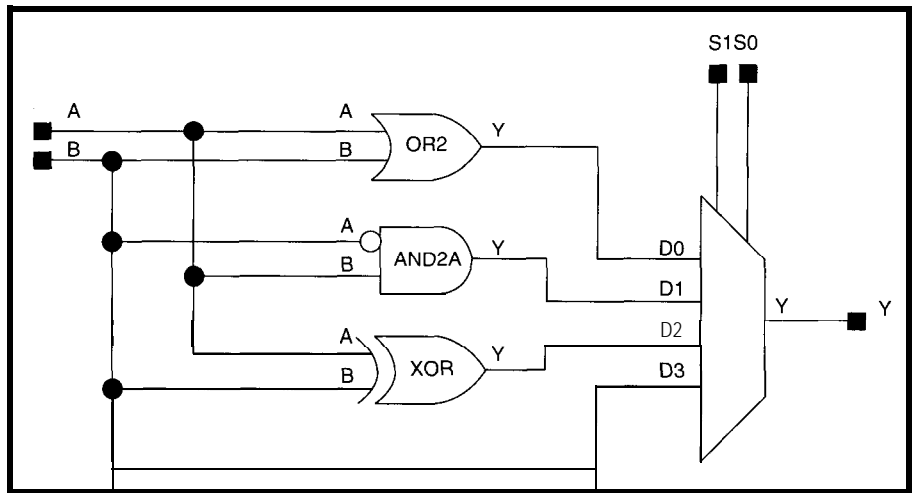
20 cells. Because latches are twice as efficient as flops, I used them for the control registers.

Actel meets the single-chip and low-power requirements, but logic mistakes cost about \$30.00 each: the chips are not reprogrammable. I used logic simulation to debug the chip's functionality before programming one. As a result, it took only three tries to get the hardware working perfectly. The remaining fifty tries were done with the simulator. Had I simulated both the memory and the LCD controller, I believe I could have saved a try or two.

The Actel design tools are like Hardware Heaven: you draw a schematic, simulate the logic, and view any and all internal nodes on a virtual logic analyzer with 1000-GHz sampling on infinite channels and infinite memory. You can place and route the chip in about 30 minutes, then analyze the postlayout timing on the same analyzer. And then, when you are satisfied with the design, you make a chip. All this can be done without ever leaving your PC. This is the way the world should be. Make your boss buy you these tools immediately.

I designed the chip using hierarchical schematic entry. I consider hierarchy to be a necessity when designing ASICs and PGAs with schematics. I try to put all the I/O pins on the top sheet and try to make it look like a block diagram. Then the details of the blocks go on different sheets on lower layers of the same design drawings.

I used Viewlogic's Viewdraw for the schematics, Viewsim for simulation, and Viewwave for looking at signals. At first, I did unit time simulations. These only consider the logic states and ignore timing. On an ASIC or a PGA, the timing simulation is only accurate after the chip is routed. Preroute timing numbers are not very useful because the interconnect delays contribute as much or even



## Hard Core System Software.

Core components of your next 80x86 embedded design, that is.

### Embedded BIOS<sup>™</sup> Embedded DOS<sup>™</sup> CodeProbe<sup>™</sup> New!

Our royalty-free Embedded BIOS product turns nonstandard hardware into PC-compatible computing engines, even 186 and V-series systems.

Now with over 95 configuration options, **full source code** (30,000 lines), and no royalties, Embedded BIOS offers the most flexibility, best control, and lowest price of any BIOS in the world.

The #1 DOS for embedded systems, our Embedded DOS operating system supports the entire MS-DOS API with full reentrancy. Its built-in high-performance microkernel supports threads, timers, semaphores, message ports and queues and offers 32,000 levels of priority in both preemptive and nonpreemptive modes. **Full source** (106,000 lines) and reasonable royalties.

Our high-performance software analyzer captures, time-stamps, and records hardware interrupts, DOS calls, BIOS interrupts, and user-defined events **in real-time** for later analysis of race conditions? interrupt activity, and service times. An absolute must-have for developers involved in asynchronous software debugging. Runs under any DOS environment with most real-time kernels.

Since 1989, we've been delivering the core software of the best 80x86 embedded designs. We've launched rockets, controlled satellites, flown in military and commercial avionics, scheduled traffic lights, rented-out cars, handled bank transactions, switched telephone calls, delivered faxes, and even worked for Uncle Sam, for starters. More and more developers are turning to General Software for BIOS and DOS solutions that offer the safety of BIOS and DOS in a high-performance real-time environment.

Call or fax for free information and a bootable product demo disk!



**GENERAL SOFTWARE** <sup>™</sup>

Tel **206.391.4285**  
Fax 206.557.0736  
BBS **206.557.4BBS**

P.O. Box 2571  
Redmond, WA 98073

**Innovative real-time system software since 1989**  
Copyright (C) 1993 General Software Inc. All rights reserved. General Software, the GS logo, Embedded DOS, Embedded BIOS, and CodeProbe are trademarks of General Software Inc.



more delay than the gates. Since most errors are in logic, not in timing, and since place and route is time consuming (30 minutes versus 2-3 minutes for a pass of unit time simulation), I used unit time simulations until I was happy with the logic. Then I used postroute timing simulation.

Design rule checking, pin definition, placing, and routing are done with Actel's own ALS tools. ALS also back-annotates the design with postroute timing delays. Then I used Viewsim and Viewwave to check the timing. First, I got the address timing and control registers working. Then I tackled the data path and host logic. Finally, the LCD timing was adjusted to match the data sheet. I also verified that the new controller chip matched the LCD timing of my existing controller board just to make sure.

I did timing analysis postlayout to accurately assess whether the chip would work. Since 98% of the timing was noncritical, I only looked closely at two areas: the memory write-pulse timing and the long-ripple-carry-chain

of the counters. I used the Actel ALS delay generator to analyze the delays from the Q-output of each flip-flop, through every possible logic path, and then the D-input of each flip-flop. This tool is excellent at answering the question, "How fast will it run?" The result was that the carry-chain was indeed the worst path. It came in at 130 ns with worst-case temperature, voltage, and process. This means the chip will clock at 7.7 MHz, well in excess of the required 2 MHz. This number could be significantly improved if I designed for speed instead of to minimize the gate count. The write-pulse timing was important because I used a gated clock to generate this signal. I wanted to verify that the write-pulse fell only after the address was stable and it rose again while the data was still stable. I verified this by looking at the waveforms generated by using the postlayout timing numbers.

#### THE APPLICATION CIRCUIT

As Figure 5 shows, the schematic of the application circuit is quite

simple. The host bus consists of an 8-bit data bus, a 3-bit address bus, a read/write control line, and a chip select. The clock inputs can be connected to any convenient source of 2 MHz ( $\pm 10\%$ ). If unavailable, a 2-MHz crystal and a few discretes can be used along with an on-chip inverter to make an oscillator. The 2-MHz clock is fed into the two clock inputs. There are two clock-input pins because the clock input on an Actel can only drive clock pins on flip-flops. I needed to use the clock through some gating logic to make the SRAM write-pulse. I used a single 8Kx8 SRAM as the video memory for a display with a resolution of 480x128 pixels. The higher address lines are for future use.

The host bus interface connects easily to a microprocessor. All read and write operations occur on the chip select (/CS) signal. For a 6800-type processor, simply gate the desired address match with E to make /CS. For Intel buses with separate write and read signals, additional gates are needed. The read/write line (R/W) is

The New  
**HCS II**  
Version 2.0

Display Time, Date &  
Variables on LCD-Link  
or Printer

Data Logging

16-bit, 4-function  
Integer Math

More Variables,  
Labels & Timers

Test Month, Day,  
Year & Day of Week

More Powerful  
Comparisons

Faster

Plus Lots More...

An upgrade consisting of a new EPROM, disk, and manual costs just \$60. Order yours today!

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

**Embedded 386sx™  
Microcomputer**

*"The PC/II is ideal for embedded PC applications."*

Only  
4" x 6"

- 386sx™ Microprocessor PC/II
- On-board VGA LCD/Video controller
- On-board Memory, ARCNET®, SCSI, Floppy Serial, Parallel & Keyboard ports

For more information please call:

(416) 245-2953

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

386sx and Megatel are trademarks of Intel Corp. and Megatel Comp. Corp. respectively

#120

#121

The Computer Applications Journal Issue #34 May 1993

41

simply low during a write and high during a read cycle. Address and data should be stable before and after /CS on a write cycle. For a read, data comes out about 50 ns after /CS goes low.

The MAX635 supplies the -13.5 volts that the LCD needs. I used a trim pot for R2 to adjust the -13.5 V. I also use -13.5 for RS-232 drivers and analog circuitry in lieu of -12 V. The 10k pot, R3, controls the LCD contrast and should be front panel mounted. This particular LCD display is quite temperature sensitive, so R3 may require frequent adjustment.

## GRAPHICS SOFTWARE

I wrote all the graphics routines as C functions. `putPix` and `putByte` are written in 68HC11 assembly language for speed. Table 3 is a list of the graphics functions.

Small characters, zoomed (large) characters, and rectangular bitblit operations are all done on byte boundaries for simplicity and speed. I have no particular need for proportional fonts. In fact, even at the

### Price and Tool Update

Actel 1010 chips are now available for under \$10 in very large (>1,000-piece) quantities. Considering that a 1010 replaces about 30-50 TTL devices or about 10-20 PALs, the pricing is quite appealing. Small-quantity prices are proportionately down. With surplus graphics LCDs in the \$10-\$25 range and current SRAM prices, the cost of adding graphics to a project can be minimal. The 1020 has 80% more gates, and other parts with much higher densities are also available.

Actel has announced a low-cost development system for their smaller devices. Logic simulation using Viewlogic is still fairly expensive, but another design option is available. Each Actel chip has four pins allocated to a "probe" function. A simple interface to a PC is available that will allow the you to select any internal node of the IC to be brought out on two "probe" pins. This way, the innards of the chip can be observed. I haven't used this myself, but Actel recommends it.

hardware level, writing a byte to the display simply ignores the three least-significant bits of the horizontal register. They are used only for single-pixel operations. Characters and blits can be written starting on any line. I chose a 7x10 font with two-line descenders. This provides one pixel between characters if the characters are on byte boundaries.

Pixel draw (`put Pix`) is in the inner loop of line draw (`lcdLine`). `LcdLine` is based on Bresenham's algorithm. It could be significantly faster if written in assembly. `LcdLine` is used by `polyLine` to draw complex shapes out of connected lines. The parameter list is a pointer to a list of X,Y points. It draws until it comes across an illegal X value.

Microsoft Flight Simulator was partly responsible for my motivation to build attractive gauges. I draw them on an X-Windows system using the `bitmap` command of the X-Windows icon editor. It conveniently outputs a static data array in C source format that can be simply included in your source. It even has `#defines` for the X and Y sizes that I simply pass to `putRect`. It couldn't be much simpler. I guess a converter could be written to convert the output of any available drawing tool that outputs bit-mapped data. Remember that the least-significant bit is the leftmost pixel of the display.

## CONCLUSION

Building the LCD controller and my application was great fun. I feel satisfied that I met all my goals. The power dissipation measured 80 mW for the LCD panel and another 80 mW for the controller chip and the SRAM. That's about the same as one low-power bipolar PAL. The drawing performance is fast enough for my application. Most of the screen, consisting of 24x30 pixel zoomed

**AVOCET SYSTEMS, INC.**

<b>Intel</b> 8051 8048 8085 8096  <b>Hitachi</b> 64180 H8  <b>Western</b> 65816  <b>Rockwell</b> 6500 6502  <b>RCA</b> 1802	<b>C and Pascal Compilers</b>  <b>Simulators</b>  <b>Assemblers</b>  <b>In-Circuit Emulators</b>  <b>EPROM Programmers</b>	<b>Motorola</b> 68000 68300 68HC11 6800 6801 6802 6803 6804 6805 6809  <b>Zilog</b> Z8 Z80 Z180  <b>TI</b> 7010 7010
---	--	---

**Source For Quality Embedded Systems**

Street, P.O. Box 490, Rockport, ME 04856  
 outside U.S., call (207) 236-9055  
 Avocet CL - FAX: (207) 236-6713

**Call today:**  
**1-800-448-8500**

characters plus three lines of small characters, is updated once per second, with a gauge needle and two moving bar indicators updated twice per second. All this is done with a 2-MHz 68HC11 which is also busy collecting

and processing data. After getting the first controller chip working, I was ultimately able to squeeze the design into the lower-cost Actel A1010. This required that I reduce away some unnecessary logic and make some

speed versus gate count tradeoffs. Since most other LCDs have simpler interfaces, controllers for them should also easily fit in the smaller device. I look forward to designing another chip to support another LCD for another project. That will be the final test of my portability goals.

My current application is 28K and growing, including a font table, icons, the graphics library, and the floating-point math library. □

*Dave Erickson is V.P. of Hardware Engineering at Datacube, where 10 MHz is considered DC.*

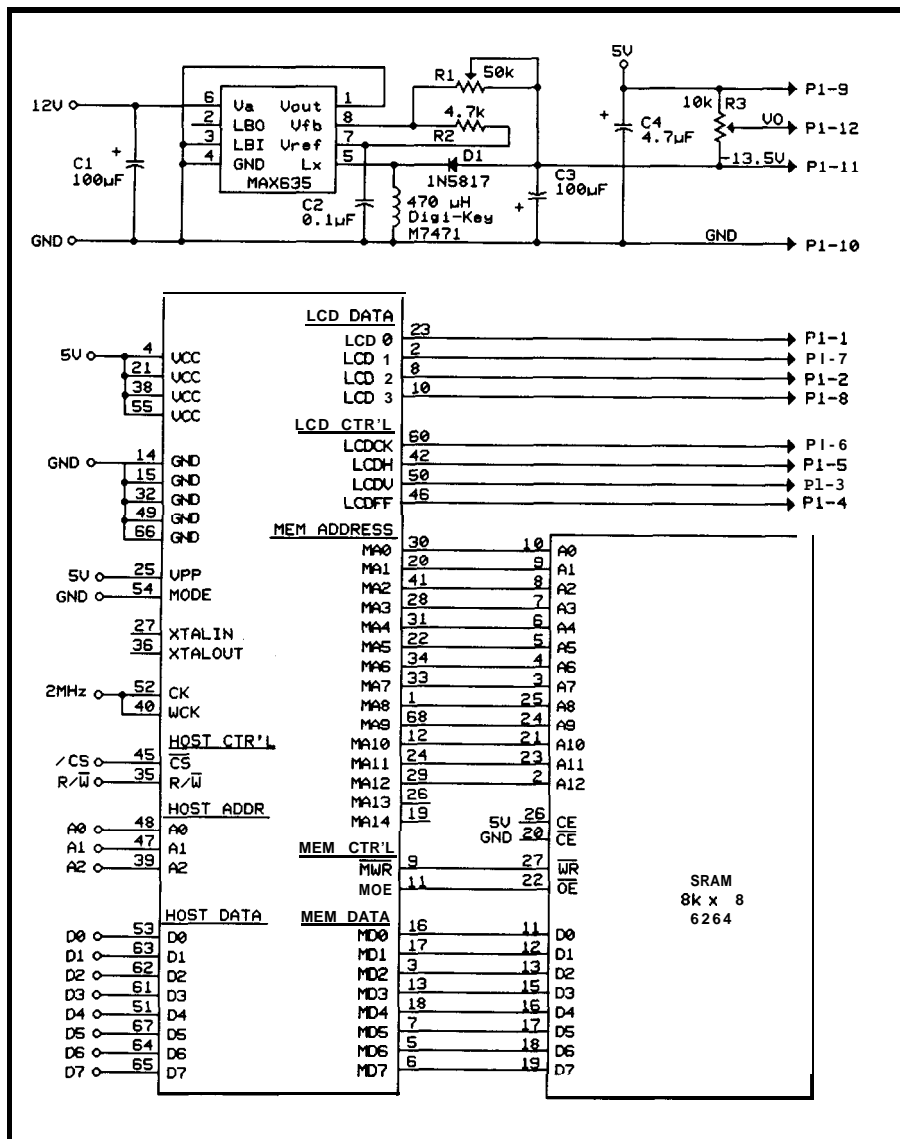


Figure 5—Since the LCD1 controller chip was designed specifically to connect directly to a microprocessor bus and an LCD display, additional components are kept to a minimum.

```

lcdByte(x, y, i)          /* Writes a byte to the LCD at x,y */
putPix(x, y)             /* Puts a pixel at x,y */
setGrMode(mode)         /* Sets the drawing mode */
lcdLine(startx, starty, endx, endy) /* Draws a line */
polyLine(list, x, y)     /* Polyline beginning at x,y */
lcdChar(i,x,y)           /* Put a character at x,y */
lcdString(cp, x, y)      /* Output a string to the LCD at x,y */
bigChar(ch,x,y, z)       /* Put a zoomed character (z) at x,y */
bigString(cp, x, y, z)   /* Output a zoomed string at x,y */
putRect(ptr, x, y, xsiz, ysiz) /* Write a rect to the LCD */

```

Table 3-A small library of C routines is all that's needed to develop applications that use the LCD display controller.

## SOURCES

The schematic, symbol, and Actel files for the chip, plus the source code for the graphics library are all available on the Circuit Cellar BBS. So are Epson-80 printable schematic files for the chip for those interested in the details. The schematic files can be used to change the design or to retarget it.

I will supply LCD controller chips (plus a terse data sheet) to interested parties for \$35.00 each. Send a check to: David Erickson, 6 Oak Drive, Topsfield, MA 01983.

The chip design is copyrighted. I ask that users treat it like freeware. You may make copies of the chip, modify it in any way you like, or design it into your embedded control product for resale. I only ask that you do not sell the chip alone or sell an LCD controller board based on it without my permission.

Actel Corp.  
955 E. Arques Ave.  
Sunnyvale, CA 94086  
(408) 739-1010  
Fax: (408) 739-1540

## IRS

407 Very Useful  
408 Moderately Useful  
409 Not Useful

## DEPARTMENTS

44 Firmware Furnace

54 From the Bench

60 Silicon Update

66 Embedded Techniques

78 Patent Talk

85 ConneCTime

## FIRMWARE FURNACE

Ed Nisley

# I/O Time: The '386SX Project Gains a Timer

A good timebase is essential in any real-time system. While the typical PC already has timers, it's helpful to have more that can be dedicated to the task at hand. Follow Ed as he adds some timers to his project.

**a**dmiral Grace Hopper used to hand out one-foot lengths of wire, observing that light travels about one foot in one nanosecond. Those wires (now collector's items, of course) helped her put electronic speeds on a human scale: "So *that's* what 70 ns means!"

It turns out that wired signals travel at about one-third the speed of light in vacuum, so one nanosecond is about four inches. One of the wait states I discussed last month reaches to the far wall of your domicile, a 16-bit I/O access stretches across the street, and an 8-bit access ends well down the block.

In this column, I'll add an 8-bit 8254 timer chip to the Firmware Development Board described last month, explore more ISA bus timing issues, and introduce a handy BIOS timing facility you may not have met before.

Keep Admiral Hopper's wires in mind..

### WHERE DOES TIME COME FROM?

Timing on PC-class machines has always been a problem because the facilities are only slightly above rudimentary. The system board includes three timers (in an 8254 or in

a smidge of LSI), but only one channel can generate an interrupt. Worse, that timer is tied into such diverse features as the BIOS time-of-day routine and the diskette motor turn-off delay, so tinkering with it can be hazardous to your program's health.

The AT added an MC1468 18A real-time clock to keep track of the date and time while the system power is off. The chip (or its moral equivalent in LSI) can produce periodic interrupts on IRQ 8 at rates ranging from 8192 kHz to 2 Hz. The BIOS implements an alarm clock function based on that interrupt, which will come in handy for the code this month.

While it's possible to use the PC timers in your code, they are best left for the normal BIOS services (unless, of course, you aren't using the BIOS services, in which case all the hardware is fair game). I decided to add a separate 8254 timer chip to support some upcoming projects, but we'll put the standard PC facilities to good use, too.

Figure 1 shows the three chips needed for the timer circuit. Of course, you'll also need the buffering and address decoding logic from last month, but the extra aggravation is fairly small. Photo 1 shows the prototype card with the circuitry thus far.

The 8254 timer is identical to the one used on the original AT system board and is rated for an 8-MHz clock. You may use an 82C54 CMOS version if one is available, as the specs are essentially identical. A -2 suffix indicates that the chip can handle a 10-MHz clock and has a slightly faster bus interface. A -5 suffix brands it as a 5-MHz dud that won't work in this circuit. The suffixes don't make sense, these chips were born back in the bad old days before rational "dash" numbers appeared.

Although It Would Be Nice to have a sensible clock frequency, I decided to use the 14.318MHz (23-foot) signal from the ISA bus connector. Pardon the digression, but I have to explain why these frequencies are what they are.

The PC's designers used an 8284A clock generator, which divides its crystal oscillator frequency by three to produce the 8088's CPU clock. The Color Graphics Adapter needed a 14.318-MHz signal to produce its composite video signal, so why not put a (cheap) 14.318MHz crystal on the 8284 and drive the CPU at 4.77 MHz?

The 8284 also produces a PCLK output at half the CPU clock rate, or 2.39 MHz. The PC's 8253-5 timer chip could handle that rate (it went all the

way up to a blazing 2.6 MHz), but the maximum interrupt rate would then be 27.5 ms, which was a bit peppy for a CPU that could execute perhaps 8,000 instructions in that time. Dividing PCLK by two produces 1.19 MHz, which the 8253 then divides by 64K to generate an interrupt every 54.9 ms.

The rest, as they say, is history.

When the AT came along, it would have been possible to boost the BIOS tick rate to take advantage of the new-and-improved 80286 CPU's horsepower, but too many programs depended on that 54.9-ms rate. Only a whole new operating system can change the clock rate; anything less is chained by historical necessity.

But, even though the PC's barnacles limit the system board 8254 to 1.19 MHz, we have no such restriction. The 8254 is rated for an 8-MHz clock, so I used half of an LS74 flip-flop to chop the 14.3-MHz signal down to 7.16 MHz.

Thus, each of the three timer outputs can produce rates from about 140 ns to 9.15 ms under firmware control. Admittedly, the faster rates aren't particularly useful, since even a 40-MHz 80386SX CPU can't accomplish much in 140 ns, but the improved resolution is still a Good Thing.

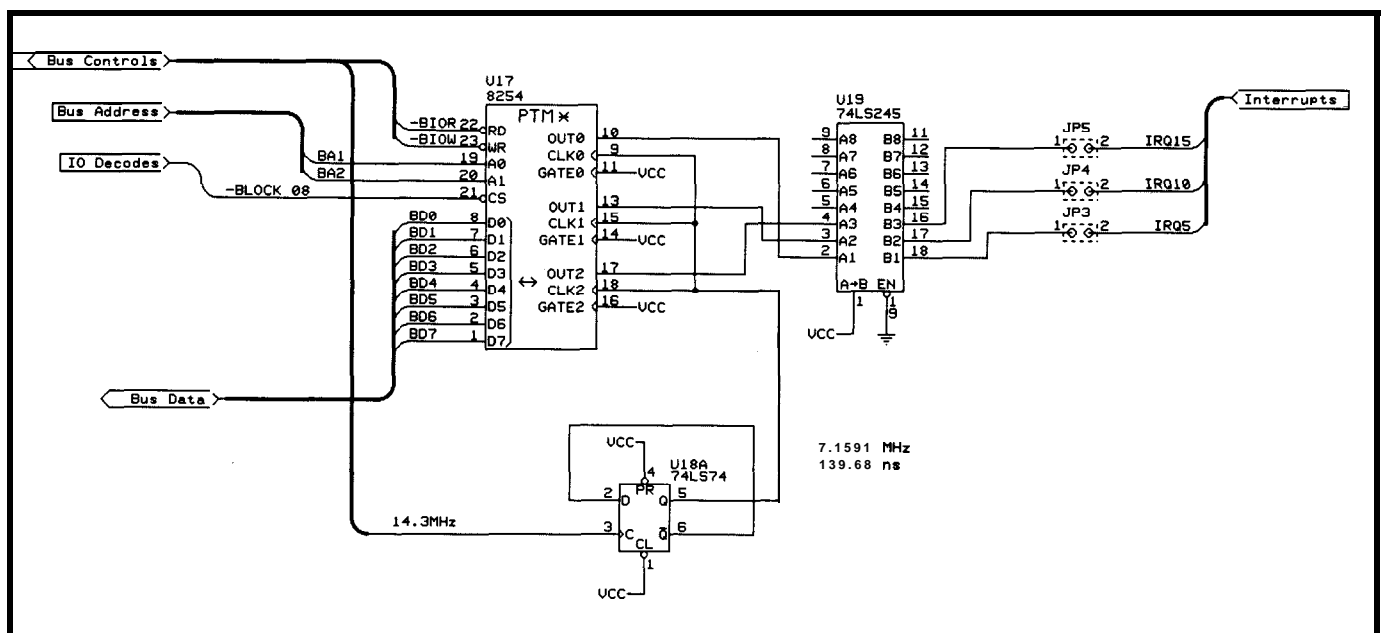


Figure 1—The 8254 timer chip shown here uses the address decoding and bus buffering circuitry presented in last month's issue. The LS245 chip drives the interrupt request lines; make sure you don't have another device connected to the same lines. In order to ensure sufficient interrupt acknowledge time, timers must use Mode 0, 1, or 3. Using 8-bit I/O accesses or additional wait states will ensure enough time for 8254 accesses. Also, if you select IRQ5 for use by the timer, be sure you don't allow LPT2 elsewhere in the system to use the same interrupt.

All three outputs drive interrupt request lines on the ISA bus connector through a LS245 which serves as a buffer. I'll assume that you're building this board for a specific reason, and that you'll build the hardware based on your requirements. With this in mind, I have made no provisions to disable the interrupts with software. If your project requires interrupts, then you'll want to install jumpers for selecting IRQ lines.

## I/O ADDRESSING

The Firmware Development Board's address decoding logic assumes

port	Read	Write
0308	Timer 0 Count/Status	Timer 0 Count
030A	Timer 1 Count/Status	Timer 1 Count
030C	Timer 2 Count/Status	Timer 2 Count
030E	nothing	Timer Control Register
031 E	Switches	LEDs

Table 1-The '386SX Firmware Development Board now has three I/O devices: sixteen LEDs, a matching set of DIP switches, and the 8254 timer. Because the address decode logic was designed for 16-bit I/O accesses, each I/O port must appear at an even address. The timer has four internal registers that are mapped into the low-order byte of four consecutive 16-bit I/O ports.

16-bit I/O operations based at even address boundaries, but you don't have to connect an I/O device to each and every data bit. For this case, the 8254 is wired to the low-order byte of the data bus and the high-order byte is not

connected. The hardware ignores the high byte during writes and the firmware must ignore the high byte during reads. Table 1 shows how devices and their ports are mapped in the address space.

The 8254 has four internal addresses that are normally accessed as successive I/O ports. On this board, however, they must be located at successive even addresses, because the decoding logic cannot handle a single-byte read or write at an odd address. As you can see from Figure 1, the 8254's A0 and A1 inputs are driven by the buffered address lines BA1 and BA2, respectively.

As I mentioned last month, the only difference between 8-bit and 16-bit I/O operations is that the expansion board activates -IOCS16 to indicate that it can handle both bytes. If -IOCS 16 remains inactive, the system board breaks 16-bit operations into a sequence of two separate 8-bit accesses by writing the low-order byte to the first address and the high-order byte to the next address.

The system board swaps the bytes around so that the bytes wind up at the correct I/O locations, but the Firmware Development Board does not include the circuitry to handle its end of this dance. As I mentioned last month, if you must support all possible I/O accesses, check the references for the details.

However, if the CPU is executing an 8-bit operation, it ignores the -IOCS 16 signal and simply writes the byte to the specified address. Because the 8254 is

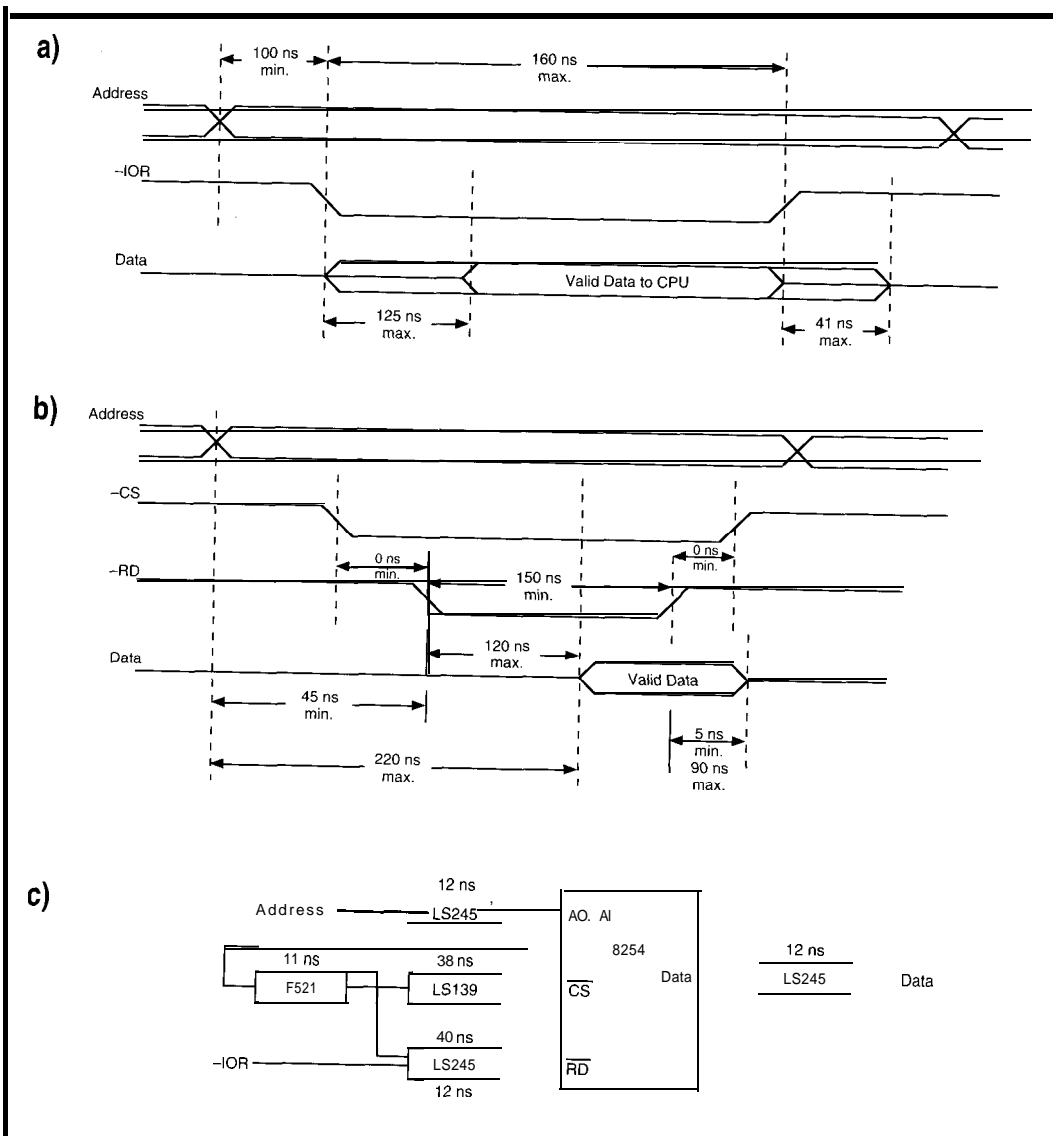


Figure 2-(a) ISA bus timings for a 16-bit I/O read operation. (b) 8254 timings for a read operation. (c) Simplified Firmware Development Board schematic showing the delay through each IC involved in a read operation. The two values shown for the LS245 in the data path correspond to the DIR and Data inputs. The delays shown in (a) must be added to the total delays along each path to find the total times.

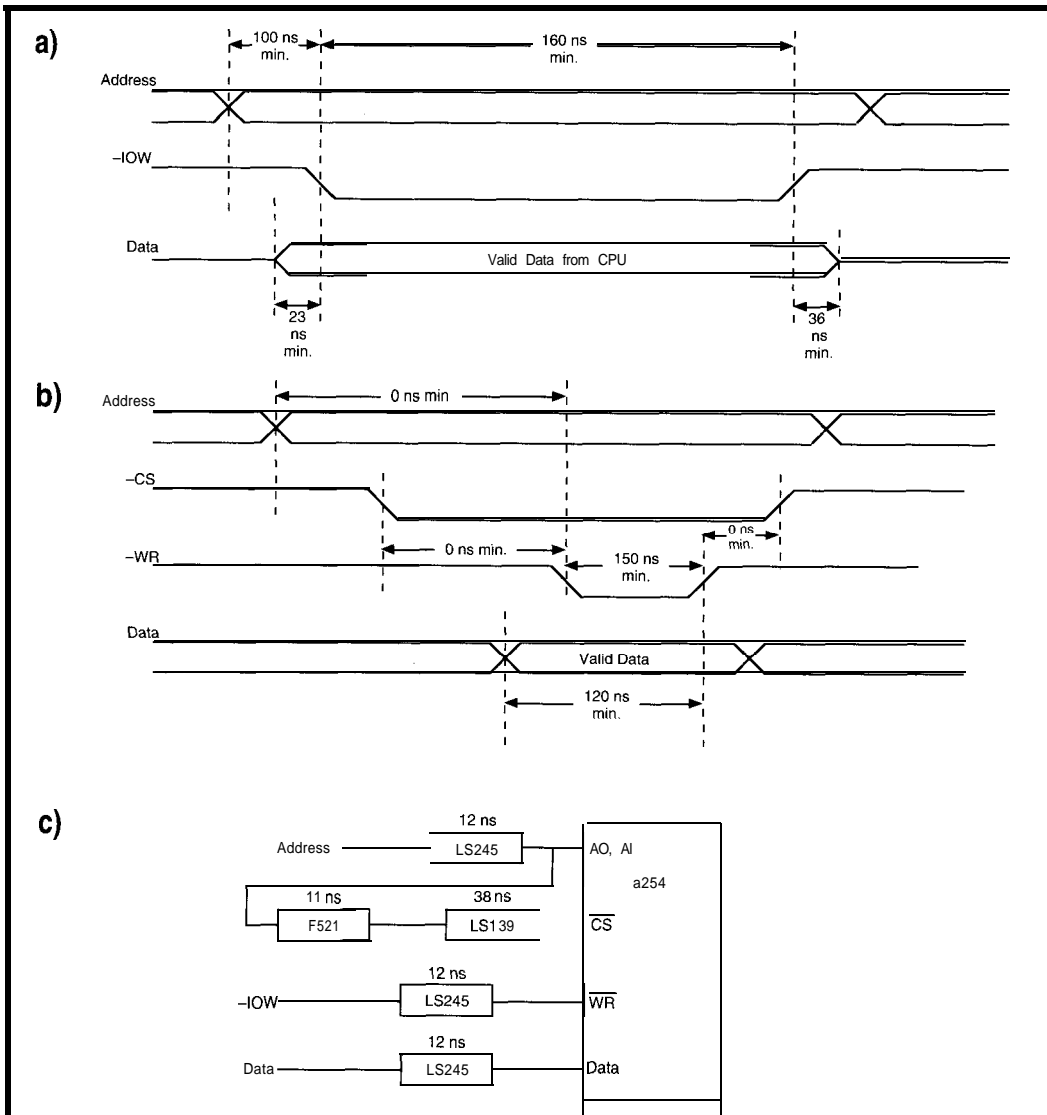


Figure 3—(a) ISA bus timings for a 16-bit I/O write operation. (b) 8254 timings for a write operation. (c) Simplified Firmware Development Board schematic showing the delay through each IC involved in a write operation. The delays shown in (a) must be added to the total delays along each path to find the total times.

connected to the low-order byte of the bus, it will respond correctly to either 16-bit or 8-bit I/O operations directed to even addresses. The high-order byte will be mush, but in this case we don't care.

The only other difference is in the bus timing, with 8-bit I/O operations requiring twice as many cycles. As it turns out, those extra cycles are critical to using the 8254.

### TIMING FROM BOTH SIDES

Because the 8254 is the first nontrivial I/O device on the Firmware Development Board, it's worth going into some of the details needed to ensure that it will actually work. This still isn't a hardware column, but if you're in the firmware business, you

must be able to interpret the hardware specs, even if it's only to point out bugs that the hardware folks get to fix!

The timing values and diagrams in this column are based on Solari's *ISA & EISA Theory and Operation*, which replaces his earlier *AT Bus Design I* mentioned in issue 3 1. The new book is \$90 from Annabooks and is essential if you're doing this stuff, but I wish it were paperbound and a lot less expensive. They do toss in a copy of their XT-AT Handbook, which is a shirt-pocket sized compendium of PC information.

Figure 2a shows the (considerably simplified) timings for a 16-bit ISA I/O bus read. Figure 2b shows the corresponding timings for the 8254's read operation. As is traditional in these

diagrams, the horizontal time axis is not drawn to scale, so you can't compare things just by looking.

According to Figure 2a, the ISA bus address settles at least 100 ns (just over 30 feet) before -IOR goes active. No more than 125 ns later, the data from the card must become valid and it must remain valid until -IOR's trailing edge. The data drivers must not remain active more than 41 ns [a mere 14 feet] after -IOR goes inactive to prevent bus contention.

As you can see in Figure 2b, the 8254's output doesn't become valid until 220 ns after the address stabilizes, or 120 ns after the beginning of the -RD pulse, whichever is later. This may look close enough, but remember that the 8254 isn't directly connected to the ISA bus!

Figure 2c shows the delay times for some of the key ICs. The LS245 buffers add a 12-ns (three

foot!) delay to the address lines on the input and another 12 ns on the data output side, so the time between a valid address and valid data on the bus is really 244 ns. Comparing that with Figure 2a tells you that this just isn't going to work.

This situation is precisely the reason why wait states were invented. Recall from last month that a single wait state adds 120 ns (about 40 feet) to the duration of -IOR, which is enough to allow the 8254's data to arrive and settle down.

There are several other paths through the logic that need checking, such as the address to -CS time versus the -IOR to -RD times. In order to be sure the chip will work, you must analyze all the paths and add up the

timings. Hint: even if you do the New *York Times* crossword in pen, use a pencil on this job!

Obviously, if you do a lot of this you'll depend on an automated timing analysis program that checks every path in your schematic using built-in delay tables for each IC. It's expensive, but better you should find problems before the board goes into production than suffer the slings and arrows of outrageous glitches in a "finished product."

As an exercise, work through the write cycle diagrams shown in Figure 3 to see if an additional 40 feet of wait state will suffice here, too.

Now, here's the punch line. If you use only 8-bit I/O operations, the ISA bus hardware will insert three extra wait states (about 120 feet) with no extra effort and no additional circuitry. Basically, by picking the right I/O operation, we can tune the system to work correctly with even this rather pokey IC!

The down side is that should you (or someone else, like the poor soul who maintains the code after you're gone) forget about this and decide to use a 16-bit I/O operation, the card will fail. Worse yet, the timings are just close enough that it's possible some of the cards will work some of the time and others won't work most of the time!

Try to find *that* bug!

However, because this column deals with sharp objects on a regular basis, I'll show you how to use the hardware either way. The checkout code this month, *timet.es.t.c*, uses either 16-bit or 8-bit I/O operations. You have control of the wait state generator so you can see just how well your hardware responds.

## COUNT THE WAYS...

Although the 8254 is probably familiar to most readers, I'll provide a capsule summary of the way it works so we will all start from the same point. A complete "how to use the 8254" is well beyond this column's charter, but, fortunately for us, we don't need all that much detail.

The 8254 contains three independent 8- or 16-bit counter/timer

The Control Word bit names are:

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

The SC bits select the Counter channel:

SC	Function
0	Counter 0
1	Counter 1
2	Counter 2
3	used for Read-Back commands

The RW bits specify the data for each counter. The counters are always 16 bits wide, but you can send one byte in special situations:

RW	Function
0	used for Counter Latch commands
1	Read/write Counter MSB only
2	...LSB only
3	Read/write LSB followed by MSB

The M bits specify the Counter's operating mode in the usual binary fashion, from 000 = Mode 0 to 101 = Mode 5. Specifying Mode 6 or 7 results in Mode 2 or 3, respectively.

The BCD bit selects the counting radix:

BCD	Radix
0	Binary, max count FFFF hex
1	BCD, max count 9999 decimal

Figure 4—Each counter channel in the 8254 is configured by an 8-bit control word written to the Control Register at address 030E. The Firmware Development Board does not support all possible 8254 configurations!

channels, each with three pins: a Clock input, a Gate control input, and an Output. Unlike 8051 counter/timers, these gizmos count down, so you load them with the number of counts you need instead of the two's

complement. If you prefer BCD to binary, you can have them count from 9999 in decades instead of from FFFF in hexits. Honest.

The three channels, referred to as Counters 0 through 2, are all 16-bit

Listing I—This Micro-C code fragment shows how to initialize all three 8254 channels to Mode 3. The *#define* macro eliminates a good deal of repetitive code and ensures that the right bytes are written in the right sequence to the right ports. Note that this code uses 8-bit I/O operations to ensure adequate bus timing.

```
#define LOADTIMERB(i, c, t)
outp(I8254_BASE+6, c); \
outp(I8254_BASE+2*i, t & 0x00ff); \
outp(I8254_BASE+2*i, t >> 8);

putstr("Timer 0 = 1 ms square wave @ pin 10\n");
putstr("Timer 1 = 2 ms square wave @ pin 13\n");
putstr("Timer 2 = 3 ms square wave @ pin 17\n");

outp(SYNC_ADDR, 0x01); /* sync at start */

LOADTIMERB(0, 0x0036, 7159); /* 1 ms @ 7.1591 MHz */
LOADTIMERB(1, 0x0076, 14318); /* 2 ms @ 7.1591 MHz */
LOADTIMERB(2, 0x00B6, 21477); /* 3 ms @ 7.1591 MHz */

outp(SYNC_ADDR, 0x00);

putstr(" . timers are now running...\n\n");
```



counters and may run in one of six different modes, known as Modes 0 through 5. Each mode uses the three pins in different ways, so you must match the firmware setup, counter mode, and external hardware in each application.

Mode 0 is a simple counter: the firmware loads a value and the 8254 decrements it by one count on the falling edge of each clock pulse, as long as the Gate control remains high. The Output goes low when the count is loaded and goes high when the count hits zero. After that, the counter continues counting but the output won't change. This mode comes in very handy for generating precise software delays to match a hardware gadget because the output can produce an interrupt with no additional hardware.

Mode 1 works the same way, but a rising edge on the Gate input serves as a trigger and starts the counter. We can't use this mode, because all the Gate inputs are tied high on the Firmware Development Board.

Mode 2 divides the clock input by the programmed count value and produces a single low pulse each time the count strikes zero. The counter reloads the initial value and continues counting, so you get one pulse out for every  $n$  Clock pulses in. This is handy for dividing a frequency by a 16-bit value, but we won't use it.

Mode 3 is similar to Mode 2 and is most useful for producing periodic interrupts. This is the mode used by the timer on the system's motherboard: the output is a square wave with a period of  $n$  clock cycles. The Gate input can synchronize the Output's phase to an external signal, but the Gate inputs will remain tied high until I can think of something that I need to synchronize with.

Modes 4 and 5 are basically Modes 0 and 1 with a twist: the Output remains high until the Count hits zero, at which point it produces a single low pulse.

If you plan to use the 8254 with anything other than a periodic Clock input, make sure you read the data sheet *carefully*. There are several gotchas that stand out only after

Are you looking for a C++ and Standard C cross compiler for DOS or UNIX, targeting 680x0, i386, i960, Mips or Sparc? Is complete source code to your compiler important to you? How about a complete Standard C library implementation, including floating point?

Hundred Acre Consulting is now offering GNU C++ and Standard C support in a number of native and cross compiler configurations. Hosts supported include MS/DOS and Windows on a 386 or 486, Sun 3 and Sun 4 with SunOS 4.1.1, RS/6000 with AIX 3.2, most System V based 386/486 UNIX systems, and many others. Call today for details!

#### Package includes:

- C++ and Standard C compilers.
- Remote source-level debugger.
- Object file support utilities.
- Complete Standard C library.
- Library of useful C++ classes.
- Binaries and complete sources.
- No royalties for redistribution.
- One year of support included.
- **Now works with Windows!**

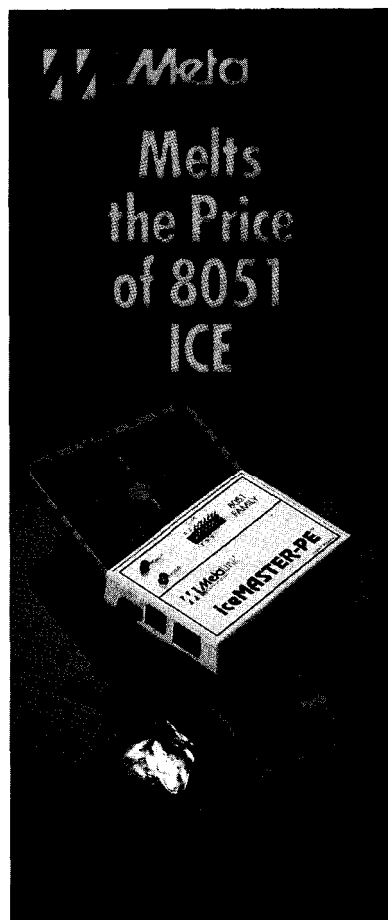
**Basic Support Package: \$495**



## Hundred Acre Consulting

1280 Terminal Way Suite 26, **Reno NV89502**  
**Phone: +1-702-329-9333 Fax: +1-702-329-9566**  
**BBS: +1-702-329-9573 Email: info@pooh.com**

#123



### Only \$851 for iceMASTER-PE

The world's most innovative emulator for members of the 8051 family is incredibly affordable. Metatink's unique Advanced Emulation Technology (AET, patent pending) delivers the best possible emulator value for engineers, consultants and students.

AET is a revolutionary design architecture that provides more features with 75% fewer components, smaller board space and lower cost. Emulator and probe electronics are integrated in a single package only 3" by 4".

MetaLink also delivers leading-edge customer service, including a 30 day money back guarantee, 10 day trial for qualified customers, rental plans and free technical support.

- Up to 40 MHz Operation
- 64K Emulation Code Memory
- 64K External Data Memory
- 128K Hardware Breakpoints
- 16K Trace Memory
- Transparent Trace (View Trace While Executing)
- Real Time & Nonintrusive
- Symbolic & Source-Level Debug
- Built-In Self-Test
- SUPPORTS 8031/8032'S
- SUPPORTS 8XC751/8XC752'S
- Windowed User Interface
- Serial link to Any PC
- Macro Cross Assembler



Call today for FREE DEMO DISK  
**(800) METAICE (800) 638-2423**  
 MetaLink ( ) ion P.O. BOX 1329 Chandler, AZ 85244-1329  
 Phone (602) 926-0797 FAX: (602) 926-1198



#124

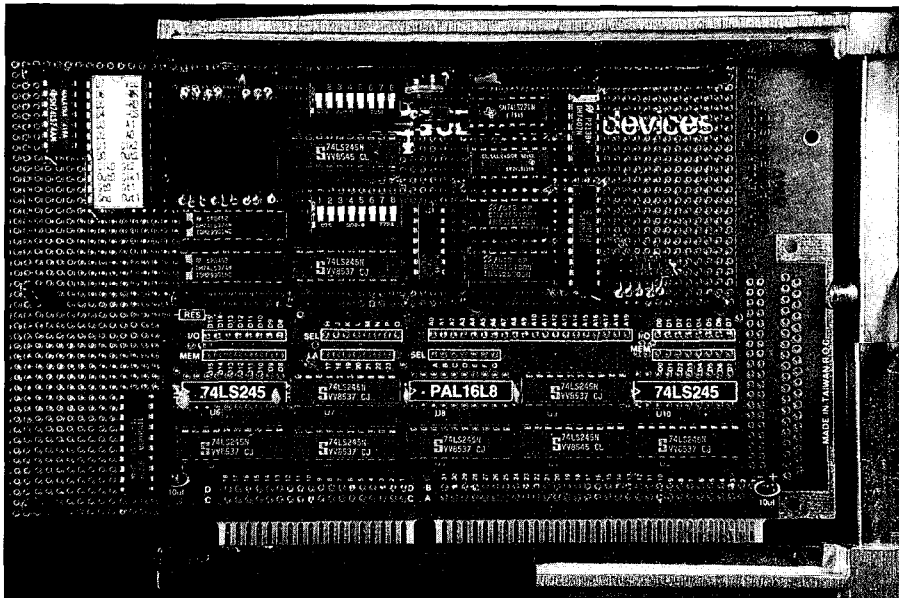


Photo 1—Addition of an 8254 timer, a dual D-type flip-flop, and an octal bus transceiver to the development board will still leave adequate space for future projects.

you've designed the thing into a circuit and it behaves, well, strangely.

Setting up a Counter channel is straightforward: write an 8-bit "Control Word" to the Control Register which is at address 030E on the Firmware Development Board. Figure 4 shows the Control Word bit layout. You must write one Control Word for each Counter channel, then write one or two bytes to the Counter channel's address (0308, 030A, 030C) to set the initial Counter value.

## HARDWARE CHECKOUT

Although in the future I may come up with other applications that require a wiring change, Mode 3 will suffice to exercise the hardware and illustrate the firmware techniques. Listing 1 shows the code needed to set up all three counters to produce square wave outputs.

The `#define` macro bottles up the three `outp()` functions needed to load the control word and write the two counter bytes. If space is tight you can convert this macro to a function call, but I opted to use this method for its simplicity.

While that code starts the timers, you'll need an oscilloscope to verify the outputs. If you don't have a 'scope, Listing 2 may be of more interest because it uses the PC to measure and display the results. If what you see on the screen is correct, the hardware is working, but if it fails, you must cadge a 'scope from a friend.

The 8254 can return a Status Byte for each counter that indicates, among other things, whether the counter's Output pin is high or low. Figures 5 and 6 show the Control Word and Status Byte values you'll need. A short routine can count the number of times the Output bit flips each second to calculate and display the output frequencies of each channel.

(Why do you write an 8-bit control word and read an 8-bit Status Byte! Because, just because!)

Remember the MC146818A I mentioned earlier? This is where it comes in handy. BIOS software Interrupt 0x15 Function 0x83 uses the periodic interrupt from that chip to set a delay; when the delay expires, the

Listing 2—The 8254 can report the state of each Counter's Output pin, so this routine can display the frequency. The code uses the BIOS alarm clock routine (shown in Listing 3) to measure out one second of real time while it checks for Counter Output transitions.

```
putstr("Timers should produce 1000, 500, and 333 counts/sec\n");
putstr(" ± one or two counts for sampling error...\n");
putstr("LEDs display iteration counter in hex\n");
putstr("Uses 8-bit I/O operations\n");

outp(SYNC_ADDR, 0x02);          /* sync at start      */

LOADTIMERB(0, 0x0036, 7159);    /* 1 ms @ 7.1591 MHz */
LOADTIMERB(1, 0x0076, 14318);  /* 2 ms @ 7.1591 MHz */
LOADTIMERB(2, 0x00B6, 21477);  /* 3 ms @ 7.1591 MHz */

Counter = 0;

while (!chkch())
  Status[0] = Status[1] = Status[2] = 0;
  Edges[0] = Edges[1] = Edges[2] = 0;
  if (SetAlarm(0x000F, 0x4240, &Alarm)) { /* alarm in 1 second */
    printf("Cannot use BIOS alarm function, code %02x\n", Alarm);
    break;
  }

  outp(SYNC_ADDR, 0x01);
  while (!Alarm)
    outp(I8254_BASE+6, 0x00EE); /* latch all status bytes */
    for (Timer=0; Timer<3; ++Timer){
      PortValue = inp(I8254_BASE+2*Timer) & 0x0080;
      Edges[Timer] += (0 != (Status[Timer] ^ PortValue));
      Status[Timer] = PortValue; /* save for next pass */
    }

  outp(SYNC_ADDR, 0x00);
  printf("\r%5u %5u %5u counts/sec", Edges[0]/2, Edges[1]/2,
        Edges[2]/2);
  outpw(BOARD_BASE+LED_OFFSET, ~ByteToSegs(Counter));
  ++Counter;

putstr("Counters continue to run...\n");
```

Bit	Function	Bit	Meaning
7	Must be 1	7	State of Output pin (0 = low, 1 = high)
6	Must be 1	6	Null Count (1 = not counting, 0 = counter loaded)
5	0 = latch current Count of selected Counter channels	5	RW1
4	0 = latch current Status of selected Counter channels	4	RW0
3	1 = select Counter 2	3	M2
2	1 = select Counter 1	2	M1
1	1 = select Counter 0	1	M0
0	Must be 0	0	BCD

**Figure 5—The 8254** includes latches that capture both bytes of the counters, so two successive 8-bit reads can return valid 16-bit data. The latches are activated by Read-Back Commands written to the Control Register at address 030E. Note that you can latch all three counters with one command. If bit 4 is zero, the first byte read from the selected Counter channels will be the Status byte.

**Figure 6—The status byte** read back from the counter channels gives you enough information to figure out what the counter is doing. Bits 5-0 echo the bits written with the counter's Control Word. In the code for this column, only bit 7 is useful.

interrupt handler flips bit 7 in the byte of your choice. This relieves your code of the need to worry about interrupts because, after you set the alarm, you simply test the target byte.

The code in Listing 2 sets up the three 8254 timers with periods of 1, 2, and 3 ms, then enters a loop that will end when you press a key. Until then, it sets up a one-second alarm and enters an inner loop that latches and reads back the current status for each

channel. The 8254 latches all three channels at once, but the Status Bytes must be fetched with three separate read operations.

Determining when each Output pin changes is a simple matter of XORing the current state with its value during the previous iteration. The Status array holds that state information for each timer, while the Edges array is incremented on each change.

Listing 3 shows the code needed to invoke the BIOS alarm clock function, which expects a four-byte delay value (measured in microseconds) and the Seg:off address of the byte to change when the delay expires. Micro-C doesn't include long (four-byte) variables, so two of the three parameters are the high and low 16-bit words of the 32-bit delay value.

When the BIOS sets bit 7 of the Alarm byte after one second, the inner loop ends. The output frequencies are just half of the Edges value for each channel because they're sampled for a full second. The code displays the results, resets the alarm, and begins accumulating another set of counts.

You'll see slight variations in the displayed totals because the counts are not synchronized to the output phases; one count may be missed or gained at the beginning and end of each second. However, if the totals aren't within a few counts of the correct values, something is badly wrong.

Just for fun, I tossed in a function to convert a byte into a pair of seven-segment hex digits. A line of code runs the iteration counter through that routine and displays the digits on the LEDs. As an exercise, adjust the code to display the digits backwards and upside-down so they read correctly from the top of the board...

## RELEASE NOTES

The code this month, *timetest*, has a variety of tests I used to get the 8254 running and verify the bus timings. You'll need a 'scope to check some of the routines, but the self-testing ones should get you on the air if you do careful wiring.

**Listing 3—The software INT 15h Function 83h** invokes the BIOS to assert hardware IRQ8 and provides an "alarm clock" for PC programs. This routine shows how to set the delay and specify the address of a byte that will change when the time expires. The code is mostly assembler inside a Micro-C wrapper.

```

/* Set up BIOS real-time alarm clock to delay interval in          */
/* microseconds. Returns 0 if OK, error code if not (also        */
/* sets *pbAlarm to return code)                                */
/* High-order bit of *pbAlarm is set when delay time expires    */
/* Remember to split the usecond count manually; 1 sec = 1E6 us */
/* The asm{} section starts with BP pushed and loaded with SP  */
                                                                    */
WORD    SetAlarm(DelayHi, DelayLow, pbAlarm);
WORD    DelayHi;
WORD    DelayLow;
BYTE    *pbAlarm;
{
    DelayHi;          /* keep compiler happy */
    DelayLow;
    pbAlarm;

asm {
    MOV  BX, SP          aim at the stack
    MOV  CX, 8[BX]      get delay high byte
    MOV  DX, 6[BX]      . .. low byte
    MOV  BX, 4[BX]      get pointer to alarm byte
    MOV  AX, DS         . .set up segment
    MOV  ES, AX
    MOV  AX, #8300      Set Event Wait Interval function
    INT  15
    JC   ?seterr        C set on error
    XOR  AL, AL         . . . clear return flag
    ?seterr MOV  BX, SP      set up for return code
    MOV  BX, 4[BX]
    MOV  ES: [BX], AL
    XOR  AH, AH        clear high byte of return value
}
}

```

One reader reported (on the BBS) that the diskette boot loader from issue 3 1 didn't work with the Award BIOS in his '486 system. I had tested the code with an old '286 AT, a '386SX laptop, and my '386 Model 80, but there's always one more system....

The problem seems to be that the Award BIOS expects the last two bytes of the boot sector to be 55AA rather than the OOED I used. It turns out that a valid hard disk partition table (at offsets 01BE through 01FD) will be followed by those marker bytes. Of course, diskettes do not have hard disk partition tables, so the marker bytes are just there.

However, the boot loader has a somewhat more serious problem. Although the code looks like it can handle files up to 64K bytes, it will fail after reading 7F00 bytes. I forgot that the diskette interface uses a 16-bit DMA controller that cannot transfer data that crosses 64K byte physical address boundaries.

The loader hit this problem when it tries to load a sector at 0800:80FF.

The corresponding physical addresses are OFF00 through 100FF, and the DMA hardware simply cannot handle the "carry" into the high-order hexit. The BIOS routine will return error 9: Attempted DMA across 64K boundary.

The quick-and-dirty fix is to change the load point from 0800:0100

to 100:0100. By starting the load at physical address 10100 the file can be up to FFO0 bytes before the address wraps. Because the binary files we are using with the loader are limited to that size, we're in good shape.

Eventually we'll build a DMA-capable interface and I'll be sure to

### Corrections

A number of mistakes crept into the Firmware Furnace schematics in the April issue (#33) during editing. Please take note of the following:

\*Page 46, Figure 1-The 16-bit extension portion of the bus should have SD15 (pin 18) connected to the bus bar with the rest of the data lines; similarly, SA18 on U1 (pin 4), BA19 on U1 (pin 15), BAEN on U1 (pin 14), and SD6 on U4 (pin 8) should all be tied to the bus bars nearest the signal names.

\*Page 46, Figure 1-On U5, signals connected to pins 4-9 should be labeled SD10-SD15.

•Page 46, Figure 1-On U5, signals connected to pins 16-1 1 should be labeled BD10-BD15.

\*Page 51, Figure 2-U1a pin 8 should be connected to ground.

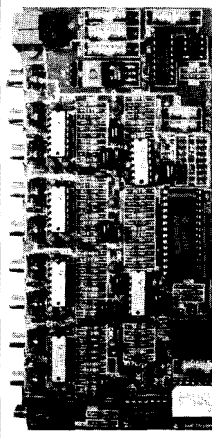
•Page 51, Figure 2-U14 pins 12, 9, 7, 5, and 3 should be labeled BA5-BA9, respectively.

\*Page 52, Figure 3-U7 and U8, pin 10 should be connected to ground and pin 20 should be connected to Vcc.

\*Page 52, Figure 3-The label above R25 should read "Left Digit (MSD)."

We regret any problems these mistakes may cause our readers.

# CIRCUIT CELLAR KITS



## HAL-4

EEG Biofeedback Brainwave Analyzer

The HAL-4 kit is a complete battery-operated 4-channel electroencephalograph (EEG) which measures a mere 6"x7". HAL is sensitive enough to even distinguish different conscious states—between concentrated mental activity and pleasant daydreaming. HAL gathers all relevant alpha, beta, and theta brainwave signals within the range of 4-20 Hz and presents it in a serial digitized format that can be easily recorded or analyzed.

HAL's operation is straightforward. It samples four channels of analog brainwave data 64 times per second and transmits this digitized data serially to a PC at 4800 bps. There, using a Fast Fourier Transform to determine frequency, amplitude, and phase components, the results are graphically displayed in real time for each side of the brain.

HAL-4 kit.....\$179.00 plus shipping

♦ The Circuit Cellar Hemispheric Activation Level detector is presented as an engineering example of the design techniques used in acquiring brainwave signals. This Hemispheric Activation Level detector is not a medically approved device, no medical claims are made for this device, and it should not be used for medical diagnostic purposes. Furthermore, safe use requires that HAL be battery operated only!

## Sonar Ranging Experimenter's Kit Targeting ♦ Ranging ♦ Machine Vision

The Circuit Cellar TI01 Ultrasonic Sonar Ranger is based on the sonar ranging circuitry from the Polaroid SX-70 camera system. The TI01 and the original SX-70 have similar performance but the TI01 Sonar Ranger requires far less support circuitry and interface hardware.

The TI01 ranging kit consists of a Polaroid 50-kHz, 300-V electrostatic transducer and ultrasonic ranging electronics board made by Texas Instruments. Sonar Ranger measures ranges of 1.2 inches to 35 feet, has a TTL output when operated on 5V, and easily connects to a parallel printer port.

TI01 Sonar Ranger kit.....\$79.00 plus shipping

## CHECK OUT THE NEW CIRCUIT CELLAR HOME CONTROL SYSTEM

- ♦ Expandable Network
- ♦ Trainable IR Interface
- ♦ Digital and Analog I/O
- ♦ Remote Displays
- ♦ X-IO Interface

**Call and ask about the NCS II**

To order the products shown or to receive a catalog, call: (203) 875-2751 or fax: (203) 872-2204

Circuit Cellar Kits • 4 Park Street • Suite 12 • Vernon, CT 06066

cover this topic in more detail so we can all get it right. Until then, remember that diskette I/O can't cross 64K byte physical memory boundaries and you'll be OK.

The downloadable files this month include a revised boot loader with the 55AA flag and a new program load address. You don't need to modify Dunfield's MON86 because it doesn't use DMA to load hex files through the serial port.

I'd hoped to get to interrupt hardware this month, but it deserves more space than I have left. Next month I'll use the new 8254 timer to explore interrupt response times and then twiddle the PC's sacred interrupt structure so IRQ 0 no longer produces INT 8... and that takes some explaining! ☹

*Ed Nisley is a Registered Professional Engineer and a member of the Computer Applications Journal's engineering staff. He specializes in finding innovative solutions to demanding technical problems.*

## SOURCES

Ed Solari's *ISA w/ EISA Theory and Operation* is available from Annabooks Fax for \$89.95 plus shipping. It replaces his earlier *AT Bus Design* book and goes into *much* more detail on the EISA specs. Annabooks is at 15010 Avenue of Science #101, San Diego, CA 92128, phones (800) 462-1042, (619) 673-0870, or fax (619) 673-1432.

Jim Mischel's *Macro Magic with Turbo Assembler* (John Wiley & Sons, ISBN 0-471-57815-0, \$39.95) suffers from a terminally cute title, but will give you the dope on an aspect of assembler programming that isn't explained elsewhere. He even builds a useful language entirely in macros; if you liked the mini-interpreter I discussed in issue 30, you'll love this. I haven't gotten all the way through the book, but it looks good so far. Available from the usual book sources; see the listing in issue 3 1.

The 8254 timer and other parts are readily available from the usual sources, but Pure Unobtainium has the bits and pieces as well as a complete schematic for the Firmware Development Board's circuitry so far. Pure Unobtainium is at 89 Burbank Road, Tolland, CT 06084-2416, phone or fax (203) 870-9304.

## 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.

## IRS

- 410 Very Useful
- 411 Moderately Useful
- 412 Not Useful

TIRED OF WAITING FOR THE PROMPT ? **DOS IN ROM!**

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

MVDISK1 64k.....\$75  
MVDISK2 360k.....\$150  
MVDISK3 1.44m. \$195

\$75

Quantity discounts!




---

\$95 EPROM PROGRAMMER

-DD ROMS FOR SBC OR MVDISK  
-PROGRAM 2764/27010 EPROMS

---

WORLDS SMALLEST PC !!!

ROBOTS ALARMS RECORDERS DOS

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

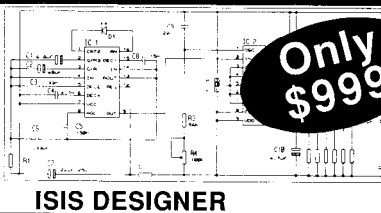
- 2 PARALLEL      -LCD INTERFACE
- 3 SERIAL        -KEYBOARD INPUT
- PC TYPE BUS    -REAL TIME CLK
- BIOS OPTION    -BATTERY OR 5V

FREE SHIPPING IN U.S.  
5 YEAR LIMITED WARRANTY

**MVS** Box 850  
Merrimeck, NH  
(508) 792 9507

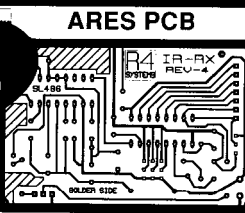


"A software system so EASY to use you will not believe it's possible."



Only \$999

ISIS DESIGNER



ARES PCB

**ISIS DESIGNER** is the Schematic Capture for *anyone* needing to enter designs. Provides netlist, multi-sheet, user configurable partslist & Electrical Rules Check report.

**ARES PCB** autorouter uses an advanced multi-strategy to achieve very high connection rates & it's fast! 10 copper layers, Design Rule Checker and MORE.

FULLY INTERGRATED SOLUTIONS STARTING AT \$999!!!

R4 SYSTEMS Inc.  
P.O.Box 451  
West Hill, Ontario  
Canada M1E 4Y9  
(416) 898-0665

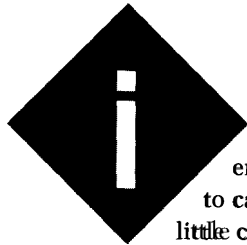
Free DEMO Package  
Write or Call Today

Download DEMO from BBS at 416 898-0508 (2400/8/N/1)

# Squeeze That Battery 'Till It's Dry

## FROM THE BENCH

Jeff Bachiochi



enjoy week-ends. It's a chance to catch up on those little chores all

homeowners enjoy. I only wish my money compounded interest as fast as the number of slips added to the job jar seem to. Nothing like a little work on Saturday to keep your mind off of the inevitable-the Sunday morning newspaper. You probably don't realize just how scary the Sunday morning paper can be. I'm not talking about the skimpy daily edition, but the multipound Sunday morning special.

Understand the world doesn't hold off its news stories until the weekend; no, the extra bulk here is not news, it's advertising. Did I hear you say, "So what?" Well, this may not cause problems at your house, but at ours, my wife, Beverly, goes over the ads with the skill of Sherlock Holmes. This isn't a problem in itself, the problem comes from a "Pavlovic" response to the word SALE. I can't tell you how much money she saves us, or how much these savings cost.

### THE BALANCE

Whether it's managing the home budget or the latest project at the office, acrobatics play an important role. Each component's effectiveness must be in balance with the additional parts count it brings, since they become a factor of the component's "cost." When a product must be portable, 25-50% of its size and weight is batteries. Squeezing every last coulomb out of those batteries is important. But, before you go bald designing for efficiency, try to select the right battery to start with.

### CUTE AS A BUTTON

The first group of cells is named after its recognizable shape. Small "button" cells are used in watches, calculators, hearing aids, singing Christmas cards, and those irritating little keychain noise makers. These cells are constructed using various materials which produce unique voltage outputs: mercuric oxide (1.35-1.4 V), silver oxide (1.5 V), lithium manganese dioxide (3 V), and zinc air (1.1-1.3 V). Discharge curves for all button cells are relatively flat over most of their operating life. Although their practical current drain rates are in the microampere range, with typical capacities in the milliamp-hour range. Larger lithium cells are being made with capacities up to 0.5 AH. These larger cells are the size of a quarter and are used mainly as battery backup devices.

The second group of cells is the most common. It consists of the familiar cylinders that power our flashlights, walkmans, and IR remote controllers. The carbon zinc, zinc chloride, and alkaline manganese dioxide construction all produce 1.5 volts. Discharge curves are sloped over the operating life of the cell, but the current-drain ratings are much higher than button cells. Current drains are in the milliampere range with upper capacities in the amp-hours.

The third group of cells is similar in size to the second, because the cells are produced to be a direct replacement for them. Nickel-cadmium cells generate 1.2 volts and are rechargeable. NiCds can support higher current drains (>1 A), but their capacities are somewhat less than their carbon-based

There are lots of kinds of batteries on the market, but which is best for what kind of application? Every manufacturer has their own data, but Jeff decided to hold his own testing session. Find out what his results were.

Number of cells	Cell voltage	Total voltage
5	1.5	7.5 volts
	1.2	6.0 volts
6	1.5	9.0 volts
	1.2	7.2 volts

Table 1-The use of series-wired batteries as unregulated inputs for linear regulators results in five NiCds not being enough for a minimum input of 6.5 V. However, the carbon zinc, zinc chloride, and alkaline types will suffice.

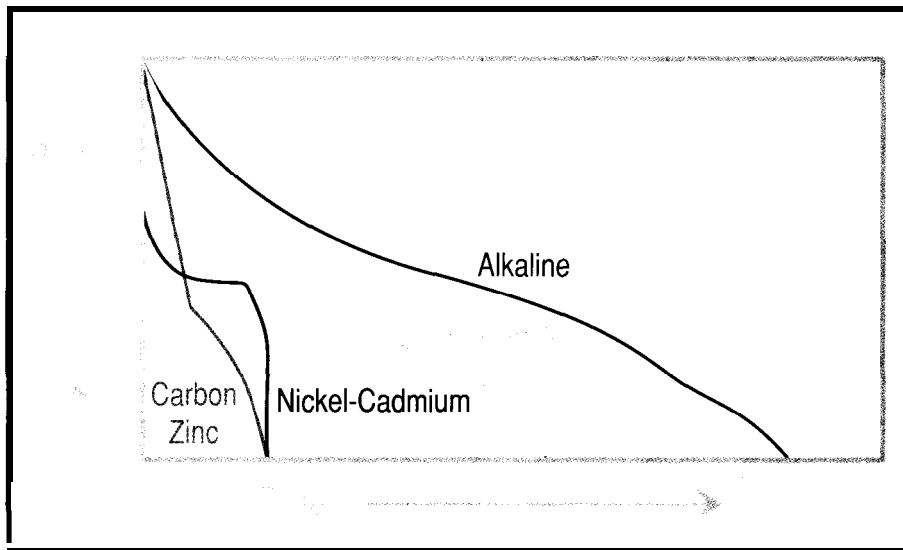


Figure 1—Based solely on time, the alkaline batteries last the longest when compared to zinc chloride, NiCd, and carbon zinc batteries.

cousins. The capability of being recharged many times offsets their higher initial costs. And, like the button cells, nickel-cadmium batteries have relatively flat discharge curves.

The final group I'll mention is the sealed lead acid cell. Sister to the automotive battery, the lead acid cell typically produces 2.3 volts (open circuit) and is packaged in groups of three or six cells to produce 6 or 12

volts. The high discharge rates and large capacities of these batteries don't occur without cost. Sealed lead acid batteries are very bulky and are not usually used in most portable equipment, but rather they are applied as backup power sources for emergency lighting and uninterruptable power supplies.

## THE GOLDILOCKS SYNDROME

Goldilocks checked the stats. "Button cells are too weak," she sighed, "and lead acid cells are just too heavy," she groaned.

"Take the carbon or NiCds, they're just right," said Mama bear, the clerk at the Bear's Country Store.

"Okay. I'll take the NiCds." Goldilocks paused, then added, "and charge 'em."

To see the stats Goldilocks referred to, take a look at Figure 1. The four curves indicate the relative discharge curves for carbon zinc, zinc chloride, alkaline, and NiCd cells (the ones you're most likely to use). These curves are given for a continuous duty load. As a battery discharges, the voltage across its known load goes down, and so does the current through the load over time. However, this is not the discharge curve a regulated battery system would have. In a regulated power supply the output voltage will be fixed. A known load will always draw a known current. As the unregulated voltage drops, the current draw will remain constant, discharging the battery at a more linear rate.

Effective battery life is a measure of the duration a battery can sustain a voltage high enough to maintain proper regulation. If we are dealing with 5-volt circuitry, the standard 78-series linear regulator typically needs a 2.0-volt drop to regulate the output correctly. Multiple cells in series can be used as unregulated inputs for linear regulators. Table 1 shows how the voltages from series-wired batteries stack up.

Five cells isn't enough for NiCds [when using 6.5 V as the minimum input voltage]. Using six cells (a nice even number) leaves a bit of head room. It's this head room that will

Listing 1—An interrupt triggers a storage sequence using ONEX1, otherwise the program checks until a key is pressed.

```

10 S = 8000H : REM NVRAM START
20 ONEX1 60 : REM JUMP ON INTERRUPT
30 G = GET : REM OTHERWISE KEY PRESSED?
40 IF G = 0 THEN 30 : REM NO LOOK AGAIN
50 GOTO 220 : REM YES JUMP TO DISPLAY
55 REM *** STORE
60 IF S > 0DFF5H THEN RET1 : REM LEAVE IF PAST END
70 FOR Z = 9 TO 2 STEP -1 : REM ALL TIME INFO
80 XBY(S) = XBY(0E030H + Z) : REM STORE A BYTE FROM CLK
90 S = S + 1
100 NEXT Z : REM ALL TIME INFO
110 XBY(0E010H) = 0 : REM CHANNEL 0 SET
120 V = XBY(0E010H) : REM READ IT
130 XBY(S) = v : REM STORE IT
140 S = S + 1
150 PRINT V, : REM PRINT IT
160 XBY(0E011H) = 0 : REM NOW CHANNEL 1
170 V = XBY(0E011H)
180 XBY(S) = V
190 S = S + 1
200 PRINT V
210 RET1 : REM RETURN FROM INTERRUPT
215 REM *** DISPLAY
220 CLEAR1 : REM STOP INTERRUPTS
230 PRINT
240 FOR Z = 8000H TO S-1 STEP 10 : REM EACH RECORD
250 FOR L = 0 TO 9 : REM ALL BYTES
260 PRINT XBY(Z + L), : REM PRINT
270 NEXT L
280 PRINT : REM NEW LINE FOR
290 NEXT Z : REM NEXT RECORD
300 GOTO 20 : REM RESTART INTERRUPT

```

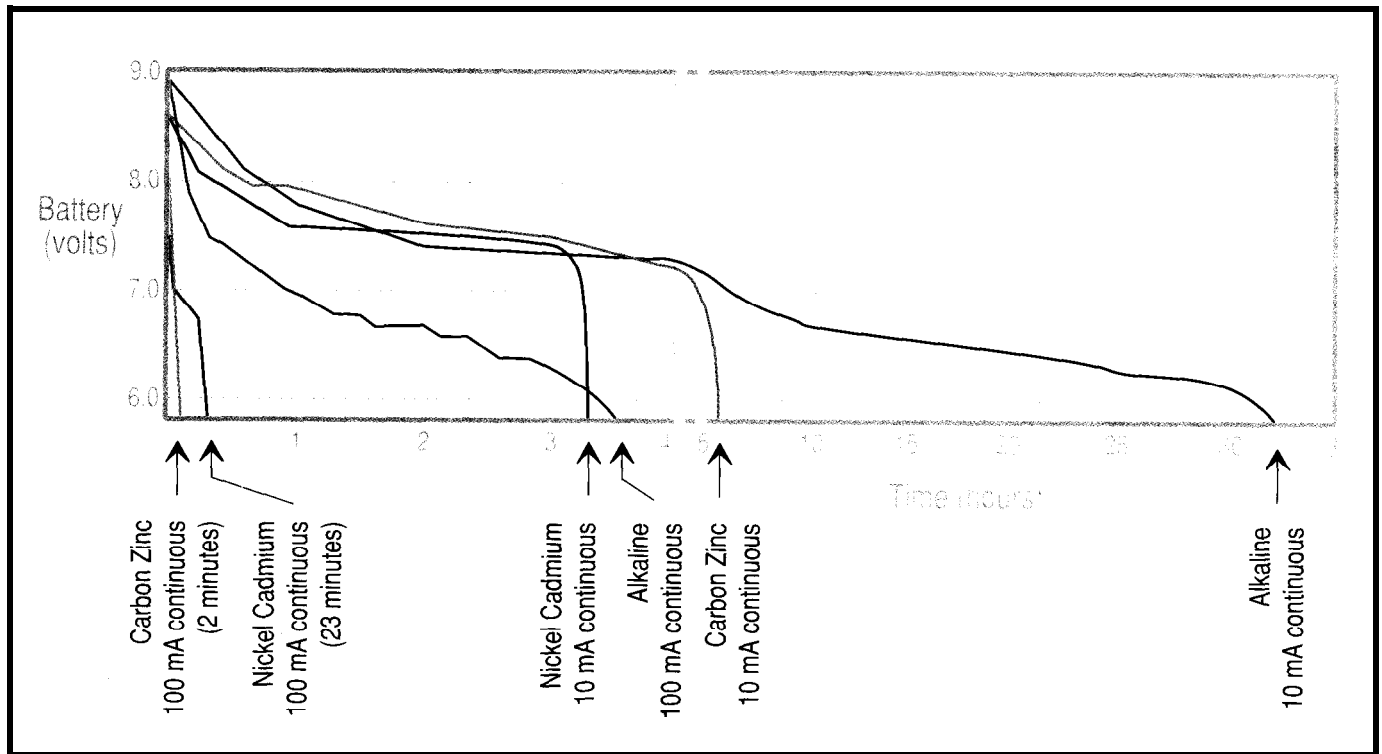


Figure 2—Most of the batteries tested with a linear regulator fared the best when current consumption was well under 100 mA. Alkaline cells lasted the longest, though other cells maintained a higher constant voltage until suddenly giving up.

determine effective battery life. (The following experiments use 9-volt batteries as their source supply. The 9-volt (7.2 for the NiCd) battery is the easiest to work with since it is a self-contained multicell device.)

### COLLECTING REAL DATA

I dusted off some circuitry that I designed back in 1989 for issue #8 to help make some observations on battery life. The RTC52 and RTC10 make great bench companions, fulfilling many of my data collection requirements. So, I hooked them up to monitor the performance of several different types of batteries. I wanted to log the battery voltage and regulated voltage over time to determine effective battery life. A simple BASIC

program loaded into the 80C52 stored a time stamp from the Oki clock/calendar along with the first two inputs from the A/D converter (which were my measurement channels during this experiment). The clock chip is set up to provide time-based interrupts (one minute apart) which trigger the storage sequence using an ON EX 1 BASIC statement (refer to Listing 1).

A simple resistor divider (100k-100k) on the battery is used to divide the battery voltage down for channel 1 since the ADC accepts a maximum input of 5 volts. Channel 2 is read directly as the regulated output for comparison.

The effective battery life for circuits using linear regulators is

shown in Figure 2. These graphs were produced from the measurements I took with my battery-loader and test jig. The carbon zinc version cannot produce a sustained current of 100 mA for more than a couple of minutes. However, under a lighter 10-mA load, battery life is extended much longer than the 10 times you might expect. The alkaline cell can sustain a 100-mA load over a period of hours, while lowering the load to 10 mA extends its serviceable life to days. Nickel cadmium gives up its complete charge in less than 1 hour (at 100 mA). At 10 mA, its life is extended by a factor of 10. This shows the NiCd's ability to supply power is independent of its current drain (to a limit).

There is an inefficiency at work here. While the battery voltage is at its highest levels, its power expenditure is also highest (remember the current through our load is constant due to the linear regulator). For example, immediately after starting a test, the battery voltage is 9 V and the current is 100 mA—that's 0.9 W out of the battery. The load, on the other hand, uses 0.5 W (5 V x 100 mA). Conversion efficiency at this point is 55% (0.5 W/0.9 W). To be most efficient, a battery's

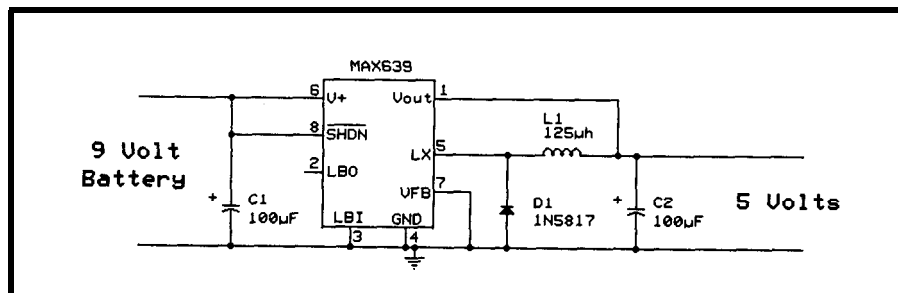


Figure 3—With the help of circuitry built for issue #8, the test setup uses a voltage divider for the ADC on channel 1, while channel 2 is read directly as the regulated output.



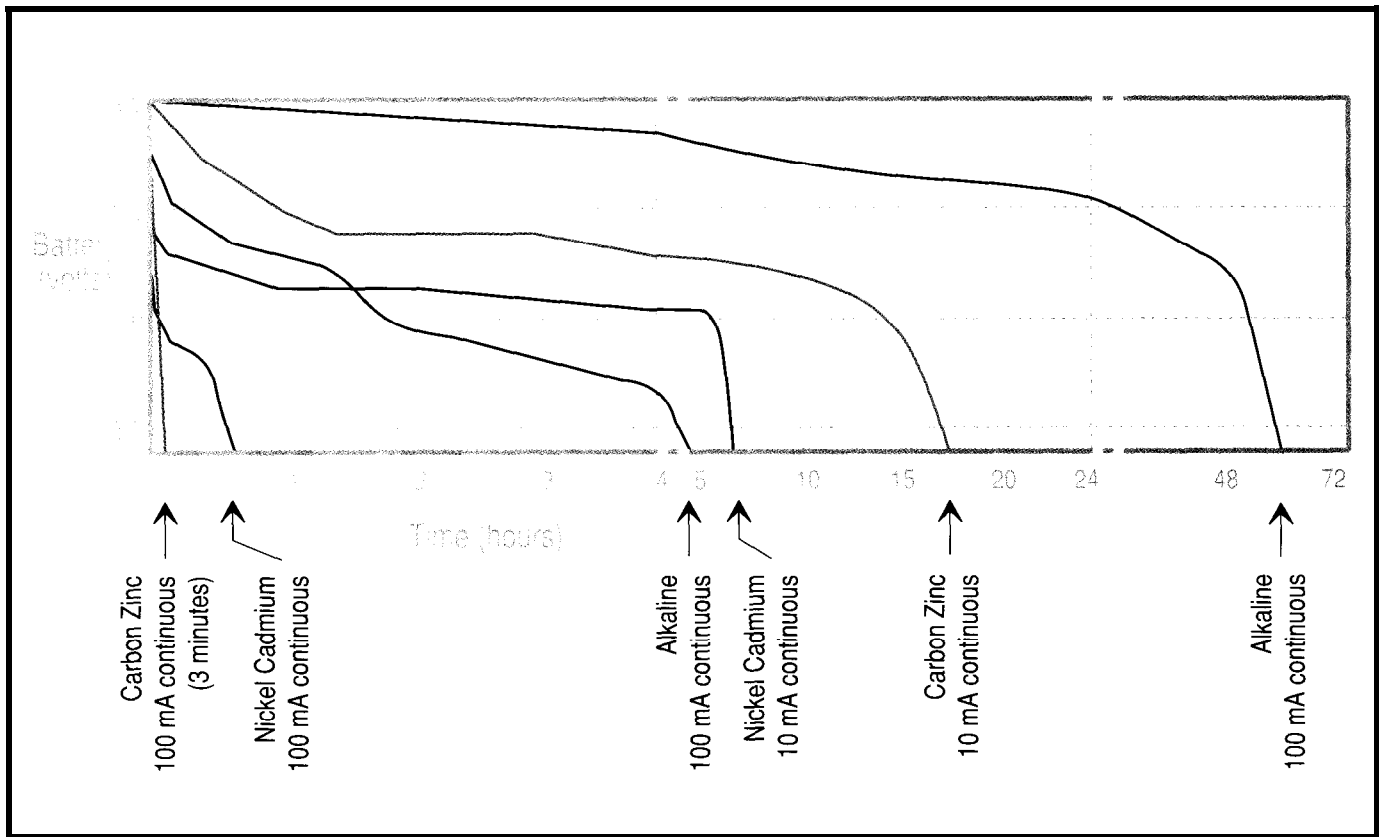


Figure 4—The optimum use of batteries is with a switching regulator. Efficiencies can improve greatly. If your next project involves the use of batteries, it would be wise to implement a switching regulator into your circuit for maximum performance.



An Official Entry Form must accompany all entries. To receive an Official Entry Form and a complete set of contest rules, write or call:

**CIRCUITCELLAR DESIGN CONTEST** **Circuit Cellar Design Contest**  
 4 Park St., Ste. 90 • Vernon, CT 06066  
 (203) 875-2199 (ask for Rose)  
 Fax: (203) 872-2204

All entries must be received by **September 17, 1993**. Prizes include \$500 for first, \$200 for second, \$100 for third, and \$50 honorable mentions.

output power should equal the power in the load throughout the entire discharge cycle.

## E FISH'N C

While the linear regulator is simple and inexpensive, unless you have tight controls on the input voltage and drop as little across the regulator as possible, they are not very efficient. A simple switching regulator can bestow efficiencies up to 90%, or so they say.

Let's find out if what they say is true. Maxim, Linear Technology, and National Semiconductor (among others) manufacture switching regulators. I chose to use a Maxim MAX639 because it is inexpensive and does not require any specially wound toroidal transformers. Only three external components are used (see Figure 3).

### Linear regulator circuitry

Input Capacitor	\$0.50
Regulator	\$0.75
Output Capacitor	\$0.50
Total	\$1.75

### Battery exchanges & cost per 100 hours (10-mA load)

Carbon Zinc	(\$0.99)	17	\$18.43
Alkaline	(\$3.49)	3	\$10.47
NiCd	(\$11.19)	31	\$11.19

### Battery exchanges & cost per 100 hours (100-mA load)

Carbon Zinc	(\$0.99)	N/A	N/A
Alkaline	(\$3.49)	28	\$97.72
NiCd	(\$11.19)	300	\$11.19

### Switching regulator circuitry

Input Capacitor	\$0.50
Coil	\$1.30
MAX639	\$3.15
Output Capacitor	\$1.50
Diode	\$0.50
Total	\$6.95

### Battery exchanges & cost per 100 hours (10-mA load)

Carbon Zinc	(\$0.99)	6	\$5.94
Alkaline	(\$3.49)	2	\$6.98
NiCd	(\$11.19)	14	\$11.19

### Battery exchanges & cost per 100 hours (100-mA load)

Carbon Zinc	(\$0.99)	N/A	N/A
Alkaline	(\$3.49)	20	\$69.80
NiCd	(\$11.19)	66	\$11.19

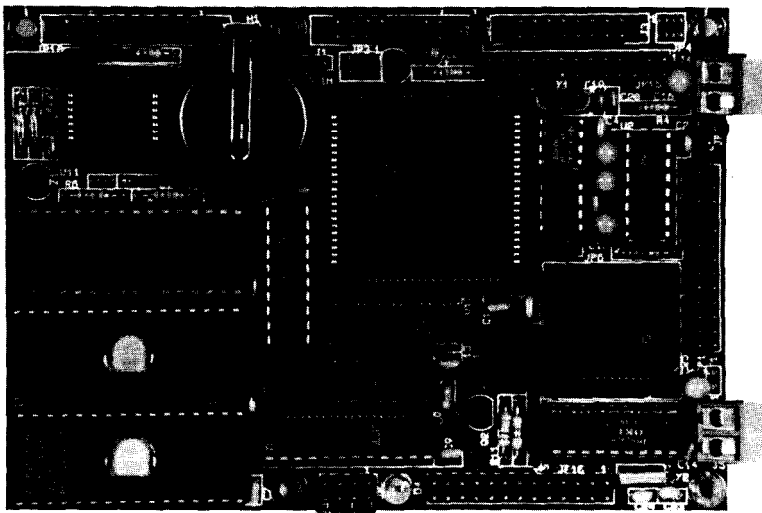
Table 2—What kind of battery and regulator circuit you use depends on the expected load and the price you're willing to pay. All prices are for 100 quantity.

The MAX639 is a DC-DC step-down converter. An internal oscillator drives a FET switch which applies the battery voltage across the coil and capacitor. When the FET opens, the coil's stored magnetic field collapses transferring its energy into the capacitor. The built-up charge on the capacitor is monitored closely by a

comparator inside the MAX639. The comparator's output is used to keep the oscillator at the ideal frequency and duty cycle necessary to maintain the desired charge on the output capacitor, while keeping the peak currents constant throughout the battery's discharge cycle, thus maximizing useful battery life.

## V25 POWER COMES TO EMBEDDED CONTROL!

Micromint's RTCV25 is the perfect marriage of an 8088-compatible processor, programming convenience, and control I/O. Forget the need for cross-assemblers and cross-compilers; your favorite high-level language for the PC and a "locate" facility are all you need. The RTCV25 enhances the V25's power with parallel I/O; A/D conversion; RS-232 and RS-485 serial ports; up to 384K of RAM and EPROM; a battery-backed clock/calendar; 128 byte EEPROM, ROM monitor, and the RTC stacking bus. Ease of code development combined with its small size and low power consumption make the RTCV25 ideal for all embedded control applications. And of course the RTCV25 is compatible with Micromint's full line of RTC peripheral boards and products.



### Features:

- 8 MHz CMOS NEC V25 processor
- Up to 384K of RAM and EPROM
- Two serial ports
- Battery-backed clock/calendar
- 128 byte EEPROM
- 40 parallel I/O lines
- 8-channel, 8- (or 10-) bit ADC
- RTC stacking bus
- Small 3.5" x 5" size
- 5-volt only operation (150mA max.)
- Full featured ROM monitor
- ROMable operating system

100 Qty.  
OEM configuration

**\$270**



**MICROMINT, INC**

4 Park St. • Vernon, CT 06066

call 1-800-635-3355

(203) 871-6170 • Fax: (203) 872-2204

The MAX639 regulates to 5 volts unless you add two resistors in the feedback circuit. But since I want a 5-volt output, I left them out. Component values for the other components in the regulator circuit are calculated with respect to the maximum operating current as shown below:

$$I_{\text{outmax}} = 100 \text{ mA}$$

$$I_{\text{peak}} = 4 \times I_{\text{outmax}} \\ = 4 \times 100 \text{ mA} \\ = 400 \text{ mA}$$

$$L = 50 / I_{\text{peak}} \\ = 50 / 400 \text{ mA} \\ = 125 \text{ mH}$$

The battery load and measurement tests are now rerun using the same prior loads. See Figure 4 for the results of this test.

The costs associated with using this component are given in Table 2. You be the judge. Questions you may ask yourself are: How often can your customer stand changing batteries? Do

they mind dealing with rechargeables? You must determine to what extent you will go in order to provide your customer the best possible product. Also, if you consider our planet's finite resources—both the raw materials needed to produce batteries and the cost of safe disposal or recycling of discharged cells—rechargeable batteries make cent\$.

### ONE FINAL CAVEAT

I made no attempt in this experiment to cover the effects of temperature or intermittent duty on battery life. All tests were made on 9-V cells at room temperature. As an alternative to the 9-volt batteries' small cell energy density, you may choose to use multiple 1.5-V cylindrical cells in your next portable project. If so, you can expect about 30% more from AAA cells, 100% more from AA cells, 500% more from C cells, and 1200% more from D cells. Your mileage may vary slightly depending on load size and battery type. Happy motoring. □

*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.*

### SOURCES

#### MAX639

Maxim Integrated Products, Inc.  
120 San Gabriel Dr.  
Sunnyvale, CA 94086  
(408) 737-7600  
MaxFax: (408) 749-6801

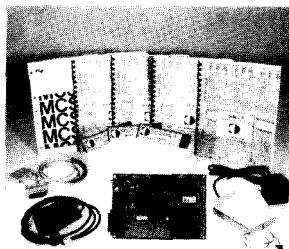
Ferrite Bobbin Coils  
Wilco Corp.  
645 1 Saguard Ct.  
Indianapolis, IN 46268  
(317) 293-9300  
Fax: (317) 293-9462

### IRS

413 Very Useful  
414 Moderately Useful  
415 Not Useful

## The \$595 Solution to 8051 System Development

# PDK51



The PDK51 is a fully integrated hardware, firmware, and software system designed to help you develop your products quickly and cost effectively.

All you need to use the PDK51 is an IBM-PC/XT/AT or compatible. We supply the rest.

**PDK51 PLUS** includes everything in the PDK51 plus Vers. 3 of our popular BXC51 805118052 BASIC compiler—\$800.

Call Now! 508-369-9556 or FAX 508-369-9549



**Binary Technology, Inc.**

P.O. Box 541 • Carlisle, MA 01741



- SBFC-II microcontroller board (8052AH-BASIC CPU) with 16K RAM
- BASIC interpreter source on disk
- SBFC-II hardware manual with schematic
- Monitor/debugger ROM plus manual
- Monitor/debugger hardcopy source listing
- UL listed 5V power supply
- Programming power supply adjustable from 4V to 26V
- KLRM11 communications program
- SXA51 8051/8052 cross assembler
- BTK52 BASIC programmers toolkit with tutorial
- 160 page BASIC Programming Manual by Systronics
- RS232 cable
- PDK51 tutorial
- Optional: aluminum case—\$45, monitor/debugger source—\$25, battery-backed real-time clock—\$39.
- All components are manufactured to exacting standards and warranted for one year.

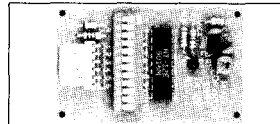


A DIVISION OF MING Engineering & Products, Inc.

17921 Rowland Street  
City of Industry, CA 91748

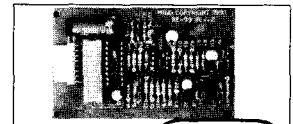
Technical Support: (818) 912-9864  
VISA  
MASTERCARD  
and DISCOVER

**1-800-669-4406**



**RC-99 RF Transmitter & Receiver Set**  
(300MHz, 12 bit address or 6 address & 4 data.)

**ONLY \$34.95**



### '93 SPRING CATALOG

**IC's:**  
DRAM memory,  
Holttek Encoder / Decoder IC's,  
Radio **Communications:**  
Motorola Spirit Radios.

#### Board-Level Items:

Digital Voice Modules,  
RF Transmitters / Receivers.

#### Security:

Auto Alarms and Accessories,  
X-10 Home **Security** Accessories  
**Infrareds.**

#### Tools:

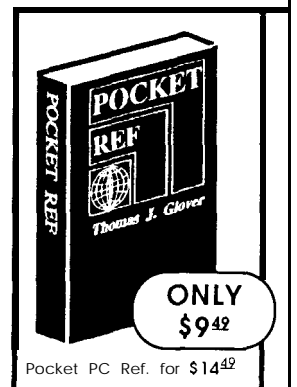
Soldering Irons,  
EPROM Erasers,  
Fluke DMM,  
IC Testers,  
Industrial PC  
EPROM Programmers,  
Ultrasonic Measuring Device

#### Prototyping PCB's:

Full line of Syntax PCB's

#### Batteries:

Alkaline  
Ni-Cad,  
Sealed **Lead Acid.**  
Button Cell.



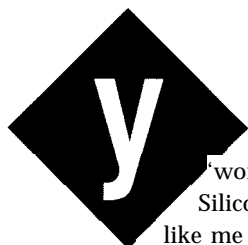
**ONLY \$9.95**

Pocket PC Ref. for \$14.95

# Audio Rx— Skippy CDs? Tangled Tapes? Call an MD...

## SILICON UPDATE

Tom Cantrell



You may be wondering why a Silicon Valley guy like me is writing an article about Sony's new Mini Disc ("MD") digital audio technology (Photo 1).

Certainly, I have no pretension to being an A/V expert. In fact, despite my immersion in MIPS, BOPS, and FLOPS, I'm quite the Luddite when it comes to the latest and greatest consumer gadgets. I'm one of the unwashed masses who can't even figure out how to make their VCR work.

Justification for an MD article goes beyond its musical merits. Rather, I'd like to highlight how this marvelous device exploits advanced technology to work its magic. As you'll see, advanced silicon—what this column is all about—plays a very big role.

For playback, the MD uses an optical read mechanism similar to that

of the CD and furthermore, the raw data-encoding and error-correction format is the same for both MD and CD. Besides exploiting proven read concepts, format compatibility means that prerecorded [i.e., "playback-only"] MDs are easily added to existing CD production lines. In both formats, pits encoding the data are formed in an aluminum reflective layer laminated to a plastic disc.

Otherwise, the MD (Figure 1) is quite different from the CD and it is these differences that define MD's compelling advantages.

Clearly, the knockout punch is MD's recordability based on magneto-optical (MO) technology. With durability and random access in MD's favor, it seems reasonable to expect it will ultimately replace tape in most applications.

With only about one quarter of the surface area of a CD, how does MD manage to store the same amount (74 minutes) of audio? The secret is an advanced data compression scheme that packs more music into each bit.

CDs are known to skip when subjected to the jarring encountered in portable or automotive applications. The MD includes 4 megabits of DRAM configured as a shock-proof memory which buffers the audio while the head gets back on track.

### MO BASICS

The MO disc combines "M" and "O" layers as shown in Figure 2. Grooves that cover the entire disc

Sony's new Mini Disc promises to revolutionize portable audio in much the same way the CD did to home stereos. Find out what makes MD so special for audio recording and what it promises in the way of data recording.

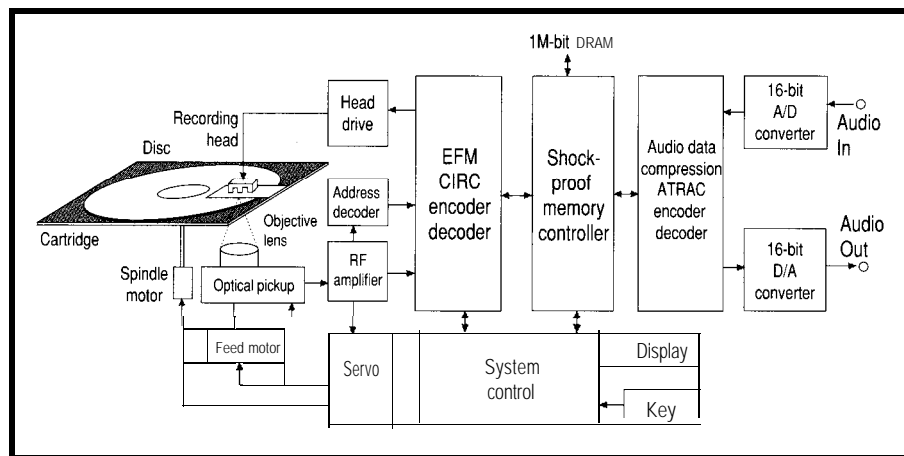


Figure 1—Just from its looks, the Mini Disc player closely resembles a typical CD Disc player. For the most part, this is the extent of the similarities between the two. One major feature that is going to be hard to resist is the capability to erase and record data and music.

cleverly encode address information (supporting random access) by “meandering” slightly. The difference in reflection depends on whether adjacent grooves have a small or large gap.

For reading, MO technology relies on a dual-function pickup that combines a laser/lens arrangement with a beam splitter and two photodetectors. When reading a read-only disc, the MD simply checks (just like a CD) for the total amount of reflected light by summing the two photodetector outputs (Figure 3a).

However, when reading a writable disc, a different technique is used in which the beam splitter funnels light to one photodetector or another depending on slight variation in polarization (Figure 3b). In this case, the difference between the two photodetector outputs determines whether a 1 or 0 is read.

The MO scheme relies on a phenomenon known as the Kerr effect, in which the polarization of the reflected light varies slightly depending on the orientation of a magnetic field.

MO writing is accomplished by polarizing the magnetic layer. The MD uses a “magnetic-field modulation overwrite” technique (Figure 4) in which laser power is boosted (from 0.5 mW for reading to 5 mW for writing). The laser heats the MO layer to a temperature (about 180°C) sufficient to dissipate any previous magnetization. At the same time, a magnetic head on the opposite side of the disc applies the newly desired magnetic field to the disc. The revolving disc cools and the applied field is retained.

Note that, in principle, MO writing could also be accomplished by modulating the laser instead of the magnetic field. Indeed, that approach is used on data MOs to allow higher capacity and speed since a laser can be switched much faster than the 100 nanoseconds or so required to achieve flux reversal with an inexpensive magnetic head.

However, laser modulation comes at a price. First, it doesn't support overwrite. Instead, either two lasers are required (one for erasing and one for recording, just like a tape) or a

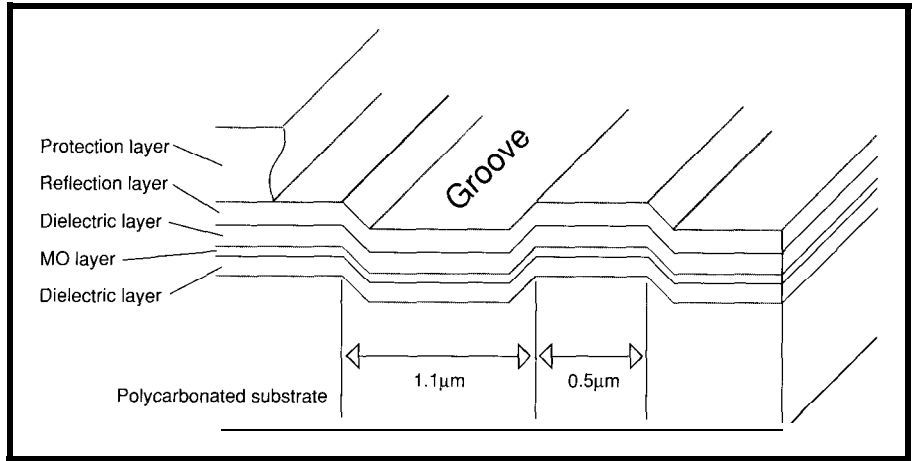


Figure 2—The Mini Disc uses a *magneto-optical* layer sandwiched between a pair of *dielectrics*. When reading a recorded signal, the reflected light is *slightly polarized* based on the polarity of the recorded magnetic field (the Kerr effect). The groove *allows* quick access to any portion of the disc even if nothing has been recorded *yet*.

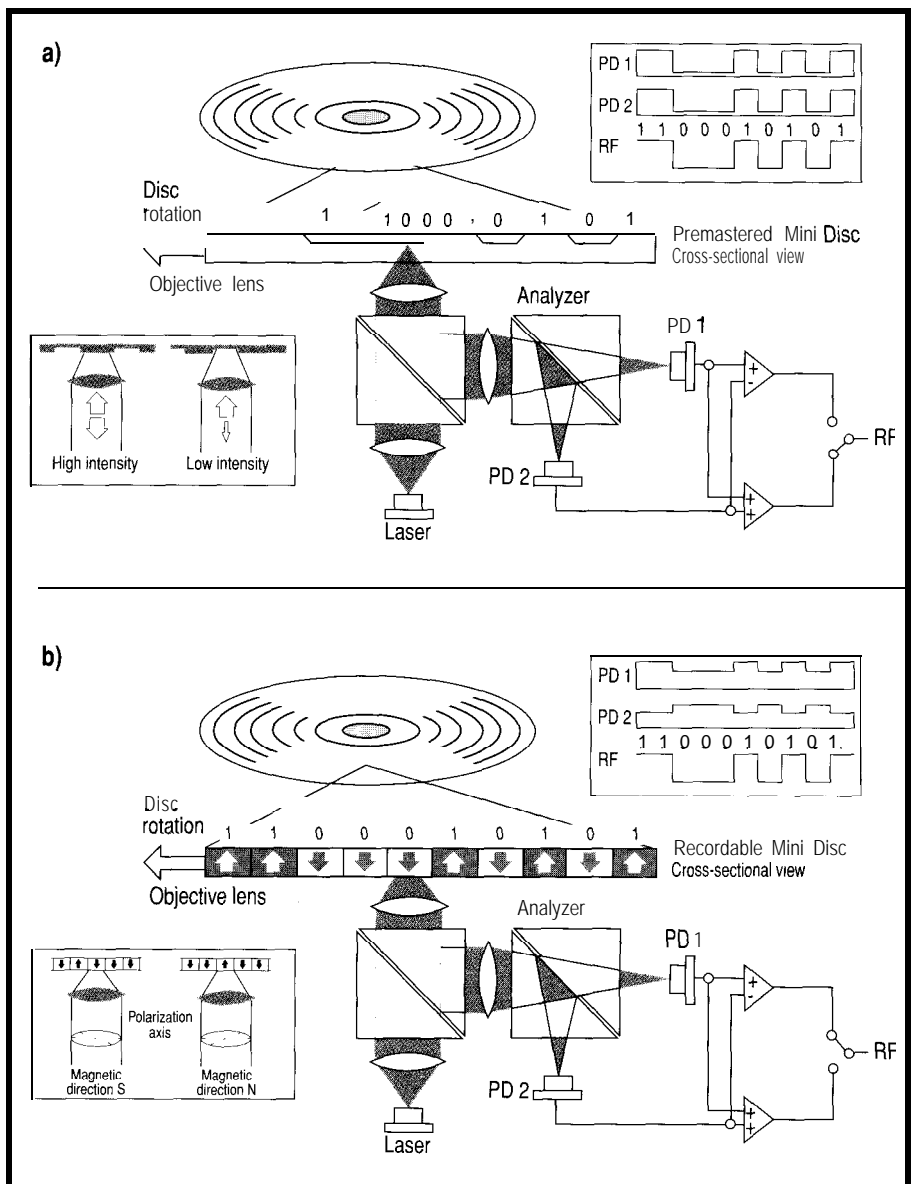


Figure 3--(a) A *prerecorded*, read-only MD has “*pits*” on the surface just like a CD and is read the same way as a CD. The total amount of reflected light is used to *reconstruct* the recorded signal. (b) When reading a *writable MD*, the difference between the polarized and *nonpolarized* reflected light is taken and used to decode ones and zeros (Note the position of the “RF” switch in both cases.)

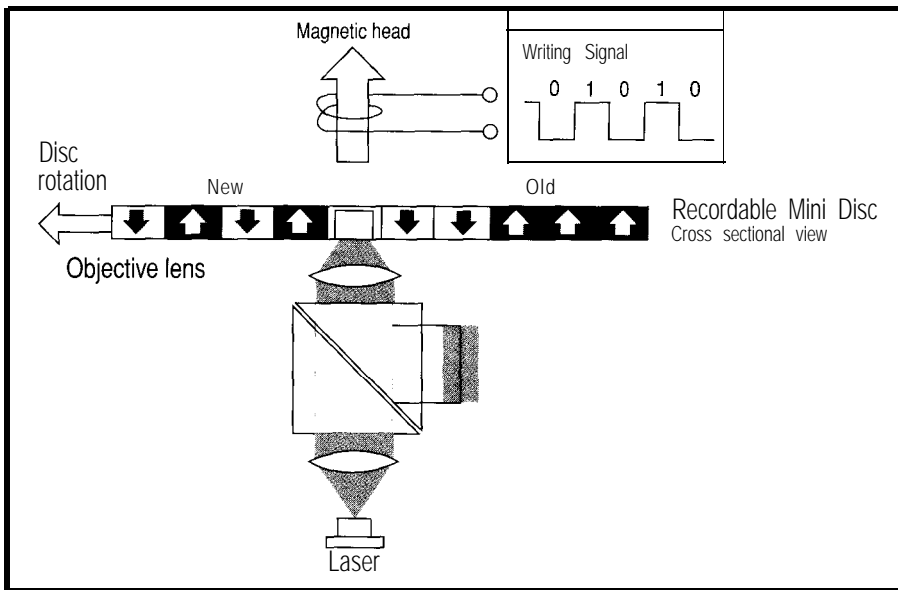


Figure 4—To write to an MD, a 5-mW laser heats up the medium from below while a magnetic write bead records ones and zeros from above. When the medium cools, the magnetic polarity is fixed. Rerecording is done in a single pass by **overwriting** the previous information, eliminating the need for an erase head or a separate erase pass.

single head must do double duty to erase and then record, complicating the design of the head assembly and spindle servos.

Also, the consistency of a laser-modulated pattern varies significantly with power, causing a problem for battery-operated recorders. By contrast, magnetic-field modulation is twice as tolerant of power fluctuations [Figure 5].

A final advantage of magnetic-field modulation is resistance to tilt during recording/playback since the

laser is only used to heat the material, not shape the recorded pattern.

The point is, given the acceptable density and speed for audio applications, the MD's lower cost, lower power, and more rugged magnetic-field modulation scheme makes good sense.

### AUDIO TRICKERY

Like a CD, the MD digitizes data at 44.1 kHz. Given 16-bit resolution and two channels, that translates into a digital data transfer rate of about 180

Kbytes per second (i.e., 44.1 x 2 channels x 2 bytes/channel) or about 10 megabytes per minute. With similar recording density but much smaller surface area, and all else being equal, the MD's tiny 64-mm disc would hold less than 15 minutes of audio.

Of course, the solution is data compression, and Sony has come up with the Adaptive Transform Acoustic Coding (ATRAC) scheme that does just that. Achieving up to 5:1 compression turns 15 minutes of bits into 74 minutes [the same as a CD] of music.

The ATRAC scheme compresses blocks comprising 512 samples (11.6 ms) using a modified discrete cosine transform (MDCT). This transform, similar to that used in MPEG and JPEG video compression standards, converts the time domain source into ordered frequency subcomponents which are amenable to further crunching. Special filtering and signal processing is applied to counteract any vagaries, such as "pre-echo," associated with the MDCT algorithm.

In particular, relying on a "psychoacoustic" model of human hearing, the various frequency values can be quantized with differing degrees of accuracy that closely reflect the ear's nonlinear response. Indeed, as shown in Figure 6, the ATRAC scheme will simply throw away source it deems inaudible.

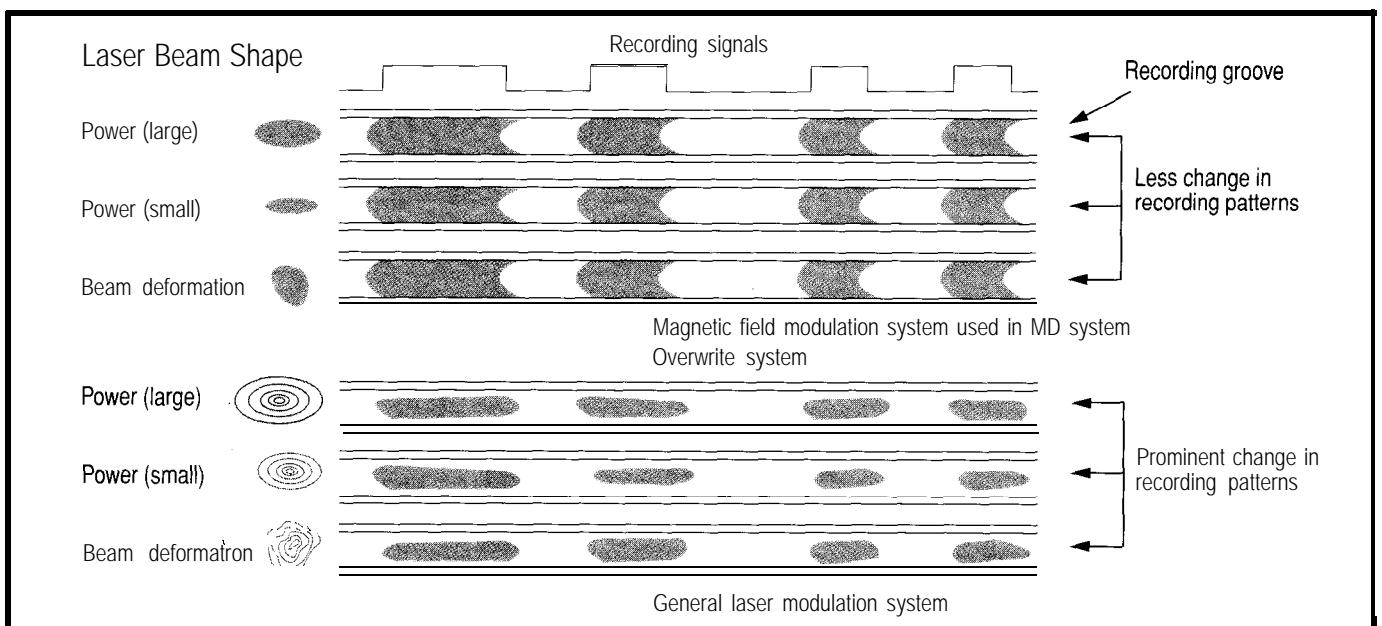


Figure 5—Using a modulated laser to accomplish magneto-optical writing is possible, however it is more costly to manufacture and the pattern recorded varies significantly with power and misaligned recording medium.



Photo 1—About the size of first-generation portable CD players, the portable MD unit allows both playback and recording in a compact package. A 1-megabit memory chip built into the playback section eliminates skipping due to shocks or bumps.

Sony faced a problem that's arguably emotional, but still pivotal: convincing the recording artists that ATRAC wouldn't compromise their artistic vision. It wouldn't do to have Michael Jackson end up sounding like one of the Chipmunks. In a classic example of "show-me marketing," Sony toured the recording studios for side-by-side sound-off challenge between MD and CD.

ATRAC isn't "perfect" in the sense that judges could often detect an audible difference between MD and CD. However, they couldn't effectively describe the difference, and most important, as often as not said MD sounded better than CD.

I'm sure the lossy aspects of ATRAC will serve as cannon fodder for endless debate about whether MD "sounds as good" as the uncompressed CD. After all, some audio gurus (known affectionately in the trade as CAGs or Consumate Audio Geeks) still argue that nothing matches the good old LP and tube amp combo. All I can say is, MD sounds just fine to me.

## MUSICAL MEMORY

There is another big advantage associated with data compression. Remember that, because the data density and linear velocity are similar, the data transfer rate for MD and CD is the same (the 180K bytes/second

demand by the analog converters). However, since MD's ATRAC system only calls for a fraction (e.g., 1/5 assuming a 5:1 compression ratio) of that, the data transfer rate required of the disc is correspondingly reduced.

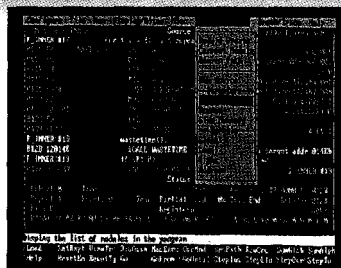
With the disc transfer rate faster than required, the MD can read ahead and store yet-to-be-played music in an on-board DRAM buffer. This is the foundation of the "shock-proof memory" which overcomes the skipping associated, most notably in cars, with CDs.

As shown in Figure 7, the MD responds to a shock by drawing down the buffer until the the head stabilizes at which point the buffer is refilled. With a 4-megabit DRAM, the MD can tolerate about 10 seconds of trouble, enough to deal with even the worst potholes.

## PIRATES BEWARE

The MD isn't the first recordable digital audio technology. That honor

# 8051 68HC11 COP8 68HC05



## iceMASTER™ Improved User Interface Features

### Rental And 10-Day Trials Available

- **iceMASTER** delivers productivity: easy to learn, easy to use and fast!
- **Hyperlinked On-line help** guides you through the emulation process.
- **iceMASTER** is FAST! The 115.2K baud serial link keeps typical download times to under 3 seconds using a standard COMM port!
- Broad support of derivative devices.
- Flexible user **interface**: you can completely configure the windows for size, content, location and color.
- **iceMASTER** is convenient! It connects easily to your PC, requires no disassembly, nor does it take up any expansion slots. It works on any PC (DOS or OS/2), Micro Channel or EISA. Even laptops!
- Supports source level debug (C and PL/M) and source level trace. 4K trace buffer with advanced searching and filtering capabilities.
- **Now virtual memory and mouse support.**



Call today for FREE DEMO DISK!  
Call today to ask about FREE 8051 Macro Assembler!  
**(800) METAICE (800) 638-2423**



MetaLink Corporation P.O. Box 1329 Chandler, Az 85244-1329 Phone: (602) 9260797 FAX: (602) 926-1198 TELEX: 4998050MTLNK

# 8051 SBC AT A NEW LOW PRICE

We are proud to offer our standard **8031SBC-10** Single Board Computer at a new, low price - just **\$79** per unit or as low as **\$49** each for quantity purchases. An 8031 with two JEDEC sockets, one RS232, 5V regulator, expansion connector. Optional second serial port, 80C31 or 32.

## Controller 80C552

At **\$149**, our 552SBC-10 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, one RS232, four JEDEC memory sockets, and more digital I/O. And we didn't stop there! You can add options like two more RS232/422/485 ports, 24 more digital I/O ports, Real-Time Clock, EEPROM, and battery-backup for clock and RAM right on board. Start with the Development board; it has all the peripherals plus a debug monitor for only **\$349**. Download and debug your code right on the SBC, then move to the OEM board above for your production needs. We also do custom design work - call for our reasonable prices.

## New 8051 Family Emulator Support

Our **DryICE Plus** product has been expanded to include support for the Siemens **80C537**. The base emulation unit is still only \$299, with the 80C537 pod priced at \$199. Other 8051 family processors supported are 8031/32, 80C31/32, 8751/52, 87C51/52, 80C154, 80C451, 80C535, 80C552/562, 80C652, and 80C51FA,B,C. Each of these pods is priced at \$149. Where else can you get an emulator with this much power and flexibility for only **\$448** - complete?

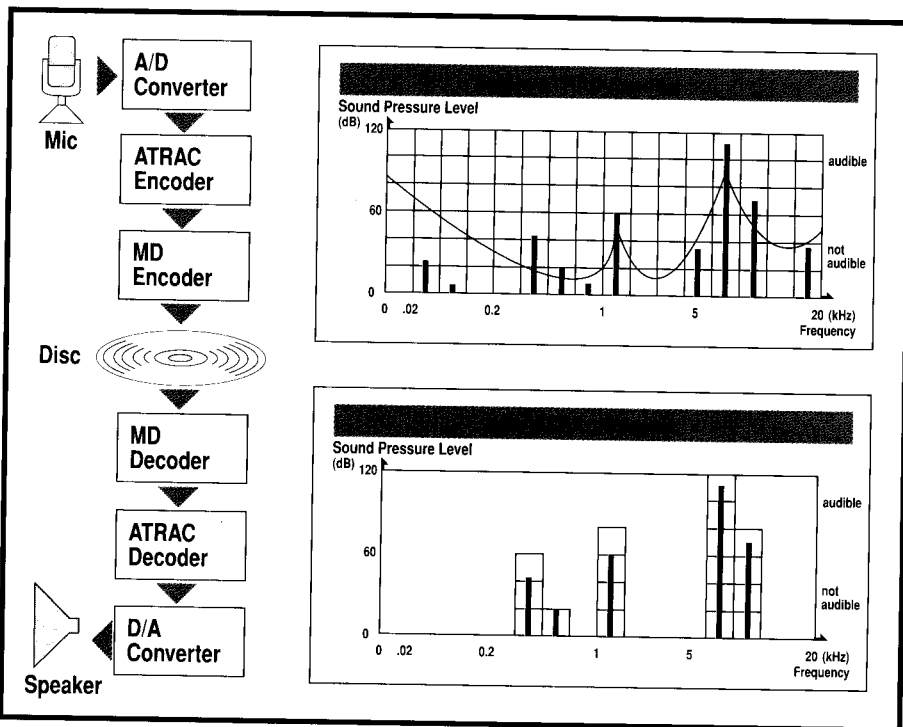
Our original stand-alone **8031 ICE** is still priced at **\$199**. Though not as flexible as the DryICE Plus, it offers excellent price/performance for learning or the occasional job need.

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



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

(619) 566-1892



**Figure 6**—In human hearing, high-level signals mask out low-level signals, making them inaudible, so the ATRAC compression system simply throws away such low-level signals. Using lossy compression similar to MPEG and JPEG video compression, ATRAC achieves a 5:1 compression ratio.

goes to DAT—Digital Audio Tape—which was developed and introduced by Sony a few years back.

It's no secret that DAT failed to take off, and the blame for that could be assigned to a variety of technical and marketing concerns. The truth is that DAT was mainly hammered by legal issues surrounding piracy, in particular, the ability to make "perfect" digital copies of CDs.

From the ashes of DAT, though too late to save that technology, have arisen a variety of antipiracy measures enabling the procession of MD with legal blessing from all concerned.

First, embedded in the hardware itself is a scheme known as the Serial Copy Management System (SCMS). In essence, SCMS allows you to make a digital copy from CD to MD but not from that copied MD to another MD. Besides bypassing both devices' analog sections, which improves fidelity, a digital copy picks up the "subdata" such as track and title info that would be missing in an analog copy.

Second, on the legal front, the "Home Recording Act" blesses the SCMS concept in consideration for royalties paid on all recording (not playback-only) equipment and media. I

don't know how much the royalty is, or to what purpose it is put (rocker retirement homes?), but it is better than nothing.

Technical and legal barriers inevitably fail in the face of basic economic incentives. The final obstacle to piracy, and the true showstopper, is the relative pricing of the R/O and R/W discs. At this time, a blank MD costs almost as much as one with your favorite sounds (e.g., \$13 vs. \$15).

While some may claim the relative pricing reflects the quality of today's music, it also cuts the bottom out of bootlegging. Pirates may be evil, but they aren't stupid enough to work for \$2/hour. Instead, they'll probably switch to bootlegging computer software.

## BEYOND TUNES

No doubt MD will have a big impact in the consumer audio world. Nevertheless, it will take time for prices to come down and a complete selection of titles to penetrate the shelf-space-constrained retail channel. While the MD means it's time to make room in the garage (next to the LPs and 8-tracks) for your audio





Photo 2—Over the years, the size of recording media has steadily shrunk while the storage capacity has steadily grown. The 120-megabyte, 2.5" Mini Disc is dwarfed by the much older, 1-megabyte 8" diskette behind it. The other popular disk sizes are 5.25" and 3.5", with typical storage capacities of 1.2 and 1.44 megabytes, respectively.

cassettes, Sony carefully states that MD is designed to complement, rather than replace, the CD. Frankly, their justifications sound rather SWAT-like (Sell What's Available Today) to me. I say the handwriting is on the wall and it says "MD Rules."

It doesn't take a rocket scientist to predict that MD also has great potential as a mass-storage device for computers. Fueled by massive consumer volume, it's likely that MD gear and media will have a significant price advantage over traditional "computer"

devices. Even at high introductory prices (\$700 drive, \$13 diskette), it's already quite competitive with existing writable technologies and, needless to say, prices won't be going up.

Also, the MD arrives just in the nick of time as computing's tried-and-true backup/interchange device—the floppy disk—rapidly runs out of gas. The problem is that software size continues to balloon with the most popular packages consuming half a dozen (or more) disks. The worst

example I've heard of is USL [now Novell/Unix], which ships 93 diskettes. Furthermore, since even pedestrian PCs sport 80–120-megabyte hard disks, the idea of backing up to floppy is more of a joke than it ever was.

Wouldn't it be nice if every PC had an MD rather than the incompatible gaggle of floppies and tapes that currently litter the landscape? No software package would require more than a single 100+ megabyte disc and one or two blank discs would handily back up all but the largest systems.

Whether it becomes the solution for better sounds or the prescription for bloated data, keep your eyes (er... eyes and ears) on MD. 📀

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

## CONTACT

Sony Corporation of America  
1 Sony Drive  
Park Ridge, NJ 07656-8003  
(201) 930-1000

## IRS

416 Very Useful  
417 Moderately Useful  
418 Not Useful

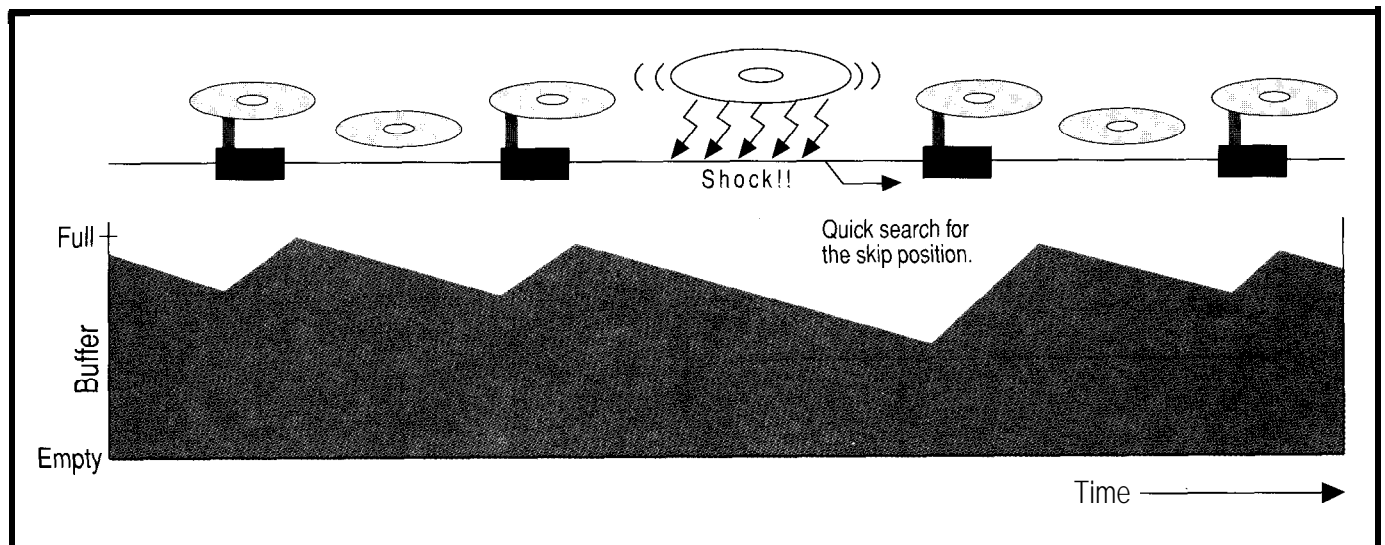


Figure 7—Unlike today's portable CD players, the MD has an on-board DRAM buffer to store music that is going to play, making it virtually immune to any type of shocks or bumps usually encountered in portable applications.

# Interchip Traffic

## EMBEDDED TECHNIQUES

John Dybowski

system constraints, and the nature of the communications traffic. Many times, the overall personality of the system ultimately imposes what is an acceptable communications solution.

For point-to-point communications between processors, you could employ standard asynchronous communications techniques using either hardware or software UARTs. The asynchronous method is not without problems since, to maintain order in the communications flow, some form of protocol is required in all but the most trivial applications. The protocol is needed to prevent data loss, to synchronize fast and slow processors, and to handle periods of real-time processing where a processor may have to drop off-line to perform some more critical operation. "Multidropping" is an available option that allows hanging multiple devices on the serial cable but usually requires the designation of a master to moderate traffic.

You would most likely have to resort to designing your own protocol since this type of low-level communications would certainly have specific needs. Remember, the crafting of a communications protocol usually amounts to no mean feat. In spite of these problems, designing a custom protocol may not be a bad way to go for some applications. However, other options exist that may be more conducive to the task at hand and it would be wise to consider the alternatives before moving ahead.

**S**erial methods of interconnecting peripheral functions and microcontrollers are

popular. Although serial buses don't have the throughput of a parallel interface, they work well when high-speed data transfer is not a requirement. Often, the reduction in interconnecting wires and the need for fewer pins offsets the slower speeds inherent in this strategy. Frequently, speed is not even an issue. Additionally, this approach is attractive, and perhaps the only choice, when working with several different single-chip controllers that do not possess a standard parallel data bus.

Maybe the first thing you would consider when selecting a serial interface is whether a synchronous or an asynchronous technique is the most appropriate. The choice, as usual, is based on the strengths and limitations of the mechanisms involved, the

What do you do when you need to add some EEPROM, an A/D converter, and a parallel I/O port to your latest design, but you only have two extra I/O pins? The InterIntegrated Circuit (I<sup>2</sup>C) bus may be your answer.

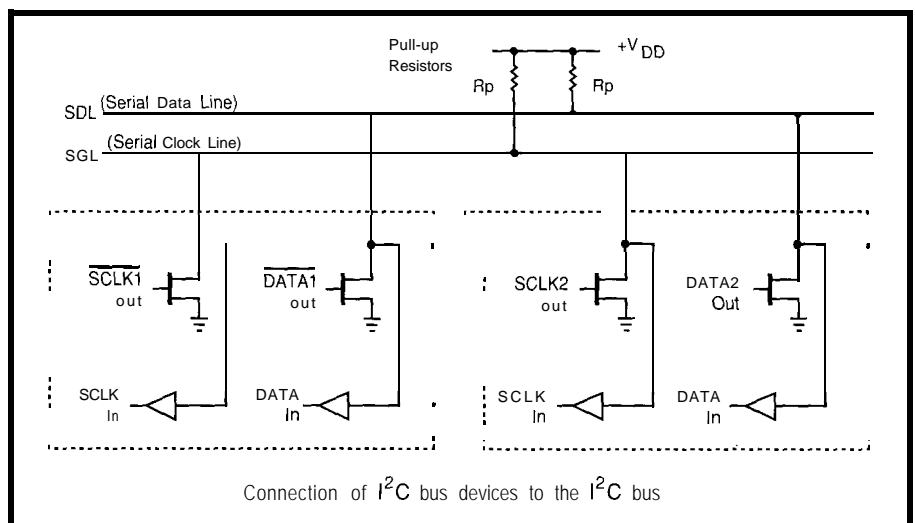


Figure 1—The I<sup>2</sup>C interface consists of a receive buffer, an open-drain transmitter, and some external pull-up resistors.

Synchronous communications schemes exist where the data transfer is performed under control of clock pulses generated by a master controller. The clock pulses signal when data bits are valid for both the sending and receiving devices. Clock-synchronized data is a good thing (especially when the clocking is done in software), since the limiting factor is the maximum clock rate. In most cases, the minimum clock frequency can go all the way down to DC. Synchronizing data transfer to a clock signal means that the controlling processor can go away and suspend communications while it attends to interrupts or to handle other timely events. The ability of a master clock to control communications alleviates the rigors of asynchronous modes where both the transmitter and receiver must endure strict timing constraints or face the prospect of losing synchronization and data.

The Motorola SPI (serial peripheral interface) is an example of a synchronous, clocked-data-transfer interface. Although it is a step in the right direction, SPI is primarily intended for point-to-point connections between a processor and a single peripheral. A multitude of options are available, presumably to allow greater flexibility in the types of devices that can be accommodated. In practical terms, all of the options ultimately make SPI sort of a nonprotocol. Because of the lack of concrete application guidelines, SPI offers little hope for any kind of standardized communications discipline, and shouldn't be applied beyond its intended purpose.

Another clocked scheme is National Semiconductor's Microwire that allows connecting multiple peripherals to a single host controller, but it suffers from a selection scheme that requires individual chip selects for each device on the bus. I've used Microwire for connecting selected peripherals to a controller. Microwire is useful when you come up short on controller pins or when the peripheral must be located off-board and you want to keep the interconnecting wires to a minimum. The fact that a lot of excellent parts are available that

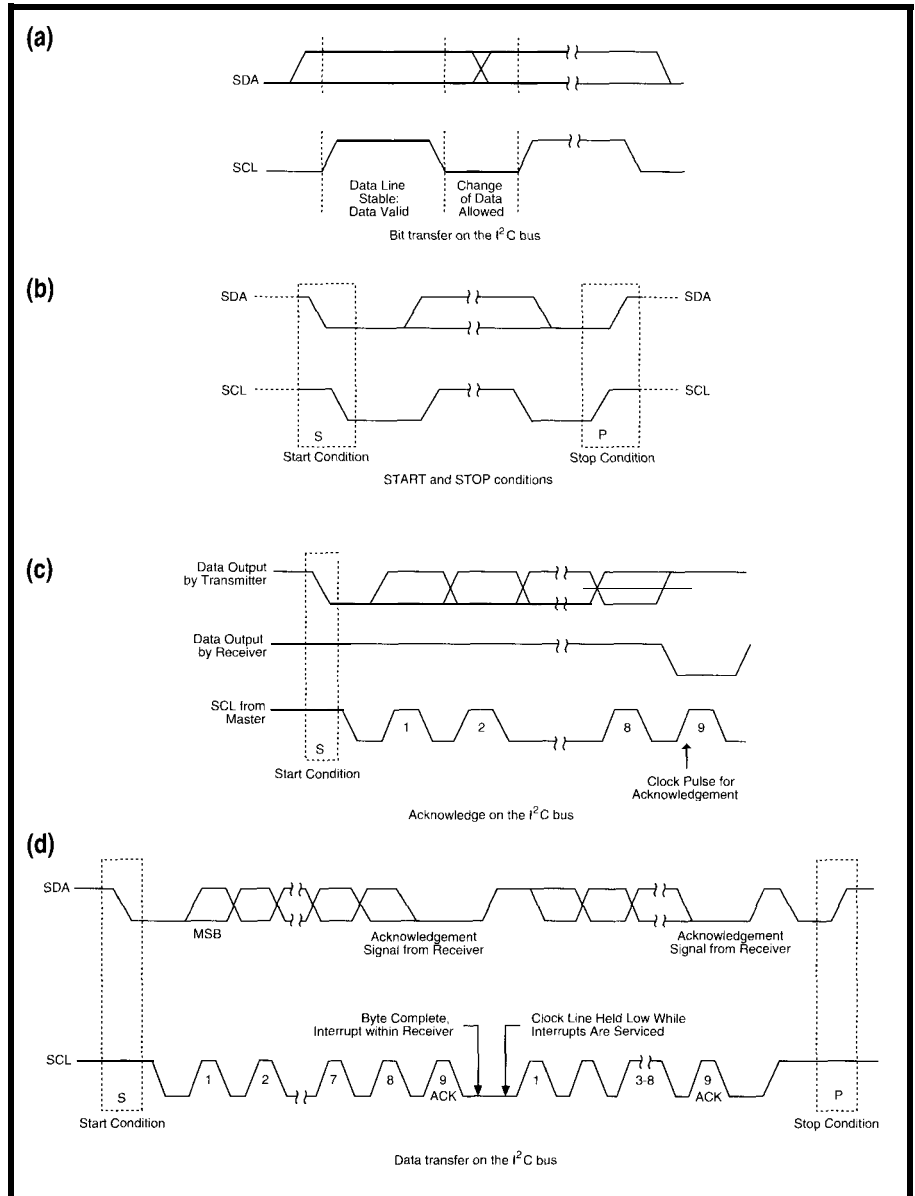


Figure 2-a) During a bit transfer, SDA must be stable when SCL is high. Changes to SDA are made while SCL is low. b) A high-to-low transition of SDA while SCL is high signals a start condition. A low-to-high transition of SDA while SCL is high signals a stop condition. c) Every byte sent is followed by an acknowledge bit sent by the receiver. d) When you put all the elements together, you have useful communication between components.

use the Microwire interface helps make it more palatable if your communications requirements aren't too stringent. Frankly, I find the need for multiple discrete chip selects on a so-called serial bus ridiculous. The additional chip select wires tend to add up rather quickly. So, I have never given Microwire serious consideration as a means of chaining together multiple chips, due to its serious limitations.

So, if you could eliminate the need to contrive a communications protocol, keep the pin count low regardless of the amount of chips on the bus, and

have access to a multitude of useful peripheral functions, you'd really be on to something. These desires are embodied in the I<sup>2</sup>C (InterIntegrated Circuit) bus. Additionally, I<sup>2</sup>C provides a simple, but effective, arbitration scheme required for resolving multimaster conflicts. This makes the bus not only suitable for hooking up peripherals, but also allows it to be applied for processor-to-processor communications.

## I<sup>2</sup>C WIRES

Electrically, the I<sup>2</sup>C interface consists of two wires that are pulled

up to a positive supply voltage with resistors. These wires are used to carry the clock (SCL) and data (SDA) signals and connect to bidirectional pins on the various bus members. Since open-collector (or open-drain) output stages are used, the lines can be simultaneously asserted by any of the devices on the bus without causing electrical problems. The bus architecture is flexible and can be used for stringing together a number of peripheral chips that operate under control of a single master controller. Multiple masters can also reside on the bus permitting direct communications between controllers, or perhaps via shared peripheral chips. Standard peripherals can transfer data at the rate of 100 kbps. The newer I<sup>2</sup>C components boost the maximum data rate to 400 kbps.

Now it may appear that two wires, a couple of resistors, and a bunch of open-drain transceivers may offer little potential towards the end of orchestrating any kind of truly flexible interface. Nothing could be further from the truth. In fact, these con-

straints turn out to be the test of an engineer's mettle. With these meager resource?, some clear thinking, and careful planning, you'll see what can be accomplished. Figure 1 shows the general interconnection scheme of the I<sup>2</sup>C bus.

The procedure of I<sup>2</sup>C communications defines unique start and stop conditions, as well as a simple acknowledgment procedure for ensuring data validity. Before proceeding, let me lay down some fundamental I<sup>2</sup>C tenets.

- \*The data on SDA must be stable during the high period of SCL.
- A high-to-low transition of SDA while SCL is high signals a start condition.
- \* A low-to-high transition of SDA while SCL is high signals a stop condition.
- \* Data is transferred in 8-bit bytes, MSB first.
- \* The bus is considered busy following the start condition.
- \* The bus is considered free a

certain time after the stop condition.

\*Every byte must be followed by an acknowledgment bit.

The bit transfer and start and stop conditions are depicted pictorially in Figure 2. Note the relationship between SCL and SDA. Also shown in this figure is the progression of a byte-wide transfer that demonstrates how the acknowledge bits are inserted by the receiving device.

In practice, SCL is used for a number of practical uses other than just clocking data around the bus. The clock can be used to empower receivers to cope with high-speed data transfers on either a bit or a byte basis. At the bit level, the I<sup>2</sup>C transfer can be slowed down when the receiver holds down SCL, effectively extending the clock low period. A device may want to cause the transmitter to wait in order to process received data or to perform some real-time processing. This hold condition can be invoked by pulling SCL low. This forces the

Everybody's using DSP ...  
... How do we get started?

IT'S **FINALLY** HERE! A COMPLETE DSP DEVELOPMENT SOLUTION BASED ON THE POWERFUL AND EASY TO PROGRAM ADSP-2111 DSP CONTROL RESOURCES PC-21 11 NOW MAKES IT EASIER THAT EVER TO BRING SPEECH TO YOUR PRODUCTS

**COMPLETE HARDWARE** The PC-21 11 contains a 12 MHz 2111 DSP, 8K Program RAM, 8K Data RAM, Microphone and Speaker Amplifiers and Expansion Port. It can also be used outside the PC with only a 5V power supply.

**COMPLETE SOFTWARE** Included software allows you to download your DSP programs and record and playback speech. An application framework and running speech recognition and speech compression code are also included. The ADSP-2111 Assembler/Linker/Simulator can optionally be purchased with the card.

**A COMPLETE SOLUTION** In short, the PC-21 11 gives you everything you need to get an immediate start on your DSP applications. If you still don't know where to start, Control Resources can quote on writing algorithms or designing custom hardware for your application. **Call or write today for more information!**

**Control Resources**

19042 San Jose Ave. Unit S, P.O. Box 8694, Rowland Heights, CA 91748  
(818) 912-5722 (VOICE) (818) 854-1439 (FAX)

# Data-Entry Terminals

## Rugged Operator Interface For Industrial Applications

FREE!  
Custom Labeling



QTERM-11  
\$235  
(7-24)

- Easy-to-read 4x20 display.
- Handheld or panel-mount.
- \*Tactile 40- or 24-key keypad.
- Programmable LEDs for status indicators and operator alert.
- Easy configuration lets you program every key with up to 255 characters.
- RS-232, RS-422 or RS-423 interface.
- Optional bar-code wand interface.
- Inexpensive batteries supply power for months. Can also be externally powered.
- FREE! Custom labeling of logo and top keys make this a part of your product.

**QSI**  
CORPORATION

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

master into a wait state until the bus is freed by the release of SCL to a high state. The clock is also used during arbitration procedures to synchronize the clocks of all transmitters in order to accomplish arbitration for bus control via SDA.

## ELECTRIC WIRES

I<sup>2</sup>C allows connecting devices with various input characteristics to the bus. These can be broadly defined as devices with fixed input levels and devices with input levels that are referenced to V<sub>dd</sub>. I<sup>2</sup>C input levels are defined in such a way that both types of inputs are accommodated. The general I<sup>2</sup>C DC electrical characteristics are specified thus:

- \*The low-level noise margin is 0.1 V<sub>dd</sub>.
- The high-level noise margin is 0.2 V<sub>dd</sub>.
- Series resistors of up to 300 ohms can be used for protection against high voltage spikes on SDA and SCL.

\*The minimum output sink current is 3 mA @ 0.4 V

\*The maximum capacitive loading for each bus line is 400 pF.

The minimum sink capabilities for output stages are specified as 3 mA and govern the minimum values for pull-up resistors. The defined low-level noise margin sets the maximum values for the series protection resistors. Bus capacitance is the total capacitance seen by the device pins and is composed of the wire capacitance and the capacitance of the connections and pins. The overall capacitance limits the maximum value of the pull-up resistors so that the signal can make transitions within the specified rise time. Refer to Figure 3 for an explanation of the DC and AC parameters for SDA and SCL.

## BUILT-IN ADDRESSING

Before proceeding, perhaps a brief explanation of some basic terms pertinent to the I<sup>2</sup>C bus would be in

order. A bus master, as the name implies, controls the transfer of data on the bus and therefore controls every slave. This is established by control of the SCL line. (This is not to imply that slave devices are excluded from asserting SCL, but that they do so only for purposes of synchronization.) The operations the master can perform include both reading from and writing to the slave devices. It follows that the master as well as the slaves can function as both receivers and transmitters.

All communications on the I<sup>2</sup>C bus are carried out over two wires. These wires carry not only the data to and from the peripherals, but they also convey the address information used to select the ICs prior to an actual data transfer. Following the start condition, a 7-bit slave address is usually sent. Recall, that transfers over the I<sup>2</sup>C bus occur in 8-bit chunks. The eighth bit, contained in the address byte, is the direction bit and signifies the direction the subsequent transfer will take. A 0 indicates a write to the slave and a 1

# The Ciarcia Design Works

**Does your big-company marketing**

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

**department come up with more ideas**

We have a team of accomplished programmers and engineers ready to

**than the engineering department can**

design products or solve tricky engineering problems Whether you

**cope with? Are you a small company**

need an on-line solution for a unique problem, a product for a startup

**that can't afford a full-time engineer-**

venture, or just experienced consulting, the Ciarcia Design Works is

**ing staff for once-in-a-while designs?**

ready to work with you. Just fax me your problem and we'll be in touch.

Remember...*a Ciarcia design works!*

Call (203) 875-2199 Fax (203) 875-8786

## µLAN

**The 9-Bit Solution is the Answer to your Embedded Network Needs**

PC/XT/AT

8051

8096

80C186EB/EC

68HC11

68HC16

Z180

**The Cimetrics Technology 9-Bit Solution is a complete microcontroller network (µLAN) that supports the 8051, 68HC11, 80186, and many other popular processors. The 9-Bit Solution takes full advantage of multiprocessor modes built into microcontroller serial ports. Our flexible software and hardware allow developers to create powerful, yet inexpensive master/slave multidrop embedded controller networks.**

- Up to 250 nodes
- 16-bit CRC error checking with sequence numbers
- Low network overhead and low resource requirements
- RS-485 interface card for the PC
- Complete source code included
- Comprehensive documentation

**CIMETRICS TECHNOLOGY**

120 West State Street, Ithaca, NY 14850

TEL: (607) 273-5715 FAX: (607) 273-5712

Characteristics of the SDA and SCL I/O stages for I<sup>2</sup>C bus devices

Parameter	Symbol	standard-mode devices		fast-mode devices		Unit
		Min.	Max.	Min.	Max.	
LOW level input voltage: fixed input levels V <sub>DD</sub> -related input levels	V <sub>IL</sub>	-0.5 -0.5	1.5 0.3V <sub>DD</sub>	-0.5 -0.5	1.5 0.3V <sub>DD</sub>	V
HIGH level input voltage: fixed input levels V <sub>DD</sub> -related input levels	V <sub>IH</sub>	3.0 0.7V <sub>DD</sub>		3.0 0.7V <sub>DD</sub>		V
Hysteresis of Schmitt trigger inputs: fixed input levels V <sub>DD</sub> -related input levels	V <sub>hys</sub>	n/a n/a	n/a n/a	0.2 0.05V <sub>DD</sub>	— —	V
Pulse width of spikes which must be suppressed by the input filter	t <sub>SP</sub>	n/a	n/a	0	50	ns
LOW level output voltage (open drain or open collector): at 3mA sink current at 6mA sink current	V <sub>OL1</sub> V <sub>OL2</sub>	0 n/a	0.4 n/a	0 0	0.4 0.6	V
Output fall time from V <sub>IH min.</sub> to V <sub>IL max.</sub> with a bus capacitance from 10 pF to 400 pF: with up to 3mA sink current at V <sub>OL1</sub> with up to 6mA sink current at V <sub>OL2</sub>	t <sub>OF</sub>	— n/a	250 n/a	20+0.1C <sub>b</sub> 20+0.1C <sub>b</sub>	250 250	ns
Input current each I/O pin with an input voltage between 0.4V and 0.9V <sub>DD max.</sub>	I <sub>i</sub>	-10	10	-10	10	μA
Capacitance for each I/O pin	C <sub>i</sub>	—	10	—	10	pF

Characteristics of the SDA and SCL bus lines for I<sup>2</sup>C bus devices

Parameter	Symbol	Standard-mode I <sup>2</sup> C bus		Fast-mode I <sup>2</sup> C bus		Unit
		Min.	Max.	Min.	Max.	
SCL clock frequency	f <sub>SCL</sub>	0	100	0	400	kHz
Bus free time between a STOP and START condition	t <sub>BUF</sub>	4.7	—	1.3	—	μs
Hold time (repeated) START condition. After this period, the first clock pulse is generated	t <sub>HD,STA</sub>	4.0	—	0.6	—	μs
LOW period of the SCL clock	t <sub>LOW</sub>	4.7	—	1.3	—	μs
HIGH period of the SCL clock	t <sub>HIGH</sub>	4.0	—	0.6	—	μs
Set-up time for a repeated START condition	t <sub>SU,STA</sub>	4.7	—	0.6	—	μs
Data hold time: for CBUS compatible masters for I <sup>2</sup> C bus devices	t <sub>HD,DAT</sub>	5.0 0	— —	— 0	— 0.9	μs μs
Data set-up time	t <sub>SU,DAT</sub>	250	—	100	—	ns
Risetime of both SDA and SCL signals	t <sub>R</sub>	—	1000	20+0.1C <sub>b</sub>	300	ns
Fall time of both SDA and SCL signals	t <sub>F</sub>	—	300	20+0.1C <sub>b</sub>	300	ns
Set-up time for STOP condition	t <sub>SU,STO</sub>	4.0	—	0.6	—	μs
Capacitive load for each bus line	C <sub>b</sub>	—	400	—	400	pF

Figure 3--A careful study of DC and AC parameters for SDA and SCL is necessary before starting your design.

denotes a read. The address byte is formatted with the actual address left justified; that is, positioned in the seven highest bits. The direction bit is contained in the LSB.

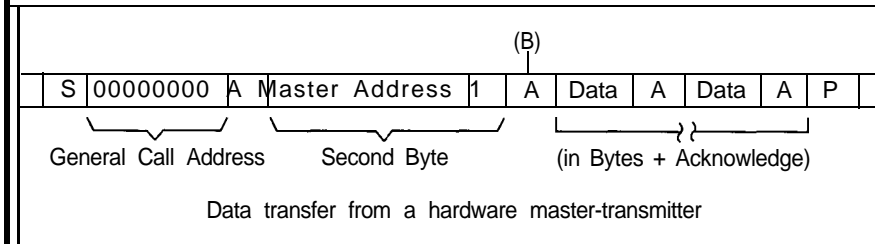
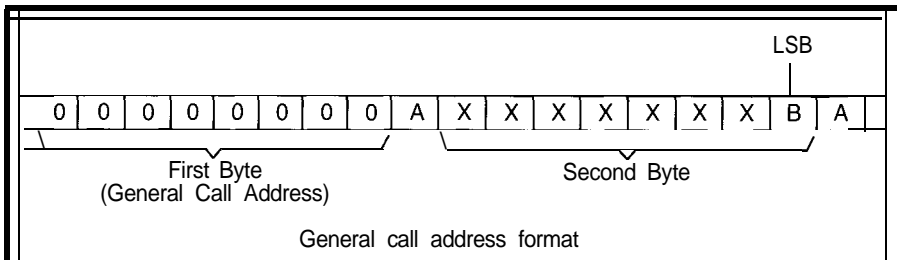
Following the recognition of the start condition, each bus member receives and checks the first seven bits of the address byte for its address. If this number matches the device's

internal address, the device configures itself for the subsequent data transfer based on the state of the direction bit.

A slave address is made up of two parts [see Figure 4). The first part is fixed and is based on the IC's general function. The fixed part of the address is the family-type base address. Its selection is coordinated by the I<sup>2</sup>C-bus committee, and is assigned based on the device's function. The programmable part of the address is selected at the IC by strapping address pins either high or low (DIP switches work nicely for this so that the address of a functional module, or assembly, can be easily set). The number of programmable address bits is dependent on the number of pins that can be assigned for this purpose. The intention of this addressing scheme is to allow multiple members of a single classification of devices to be present on the bus at the same time. For example, if you have four fixed and three programmable address bits, eight devices of the same family type could be connected to the bus simultaneously.

In addition to the specific addresses defined for the various device types, I<sup>2</sup>C defines address 0 as a special functional address. This address is referred to as the general call address. Effectively it is a broadcast command, and the issuance of this address serves to select every device connected to the I<sup>2</sup>C bus for the coming transmission. (If a device does not need the data that is sent as part of the general call sequence, it can ignore it by not acknowledging it.) The general call is a two-byte sequence and its interpretation is contained in the second byte. The meaning of this second byte is affected by its LSB. If the second byte's LSB is 0, then "00000110" is a command for the slave to reset itself and accept the programmable part of the slave address, and "00000100" is a command for the slave to accept the programmable part of the slave address without resetting itself.

If the second byte's LSB is 1, then the two-byte sequence is a hardware general call and the remaining seven bits of this byte contain the address of the hardware master itself. This could



Definition of bits in the first byte

Slave address	R/W bit	Description
0000 000	0	General call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Address reserved for different bus format
0000 011	X	Reserved for future purpose
0000 1XX	X	
1111 1XX	X	
1111 0XX	X	10-bit slave addressing

Figure 4-Each PC device on the bus has a unique network address. When the slave address specified is zero, then the packet is recognized by all slaves on the bus.

be used, for example, in the case of an intelligent device such as a keyboard which could not be programmed with the address of the slave device. In such a situation, the only recourse for the transmitter would be to generate a general call and append its own address in order to identify itself. This information would then be recognized by the system controller which would properly route the data to its final destination.

### ARBITRATION PROCEDURE

All masters generate a clock signal on SCL to coordinate the data transfer. As I mentioned, clock synchronization is performed using the wired-OR

characteristics of the SCL line. As the transfer sequence progresses, a high-to-low transition on SCL initiates the counting of the low period and once this count expires the device will allow SCL to go high. If it happens that another device is holding SCL low, the actual state of the clock line will not change. This condition is recognized by the device driving the pin high. What happens then is the device that put its SCL line high will now enter a wait state that will not terminate until the other devices driving SCL low finally release the line to a high state. Subsequently, the first device that completes its high time will again pull SCL low. This synchronization

# KELVIN

ELECTRONICS  
10 HUB DRIVE, MELVILLE, NY 11747  
(800) 645-9212 (516) 756-1750  
(516) 756-1763/FAX M/C & VISA

**100 Basic**   **150 Basic+**   **200 Advanced**   **PRO 400**  
**Standard Features - Models 100, 150, 200 & 400**  
 ● AC & DC VOLTAGES   ● DC CURRENT  
 ● RESISTANCE   ● CONTINUITY TEST-Buzzer  
 ● DIODE TEST   ● 3 1/2 Digit LCD   ● LOW BATTERY INDICATOR  
 ● ACCURACY +/- 0.5% RDG

**BATTERY TEST**   **TRANSISTOR**   **20MHz FREQ**  
**KELVIN**   **CAPACITANCE**   **COUNTER**  
 # 990097   **AC/DC**   **AC/DC CURRENT**  
**100 Basic**   **CURRENT**   **CAPACITANCE**  
**\$1995**   **KELVIN**   **LOGIC TEST**  
 1990091   **TRANSISTOR**  
**200 Advanced**   **LED TEST -**  
**TRANSISTOR**   **BATTERY TEST**   **VERIFY GOOD/BAD**  
**KELVIN**   **\$3995**   **KELVIN**  
 # 990090   **#990092**  
**150 Basic+**   **30 DAY**   **PRO 400**  
**\$2995**   **MONEY**   **\$6995**  
**BACK**

**2 Year Warranty (Parts & Labor)**

**KELVIN 94 The Ultimate Meter**  
**LCR - Hz - dBm - True RMS - Logic Probe**  
 The only meter with 0.1% Accuracy on DC Voltages, built-in True RMS, Freq Counter to 20MHz Res: 10 Hz, LCR-Inductance Tester Res: 10 uH, DC/AC Voltages Res: 0.1mV, Ohm Meter Res: 0.1 ohms

**TRUE RMS PLUS**   **12 INSTRUMENTS IN ONE -**  
 Model 94   **AC & DC VOLTMETERS,**  
 #990111   **AC & DC CURRENT, dBm,**  
**\$19995**   **OHMMETER, DIODE TESTER,**  
**AUDIBLE CONTINUITY TEST,**  
**20 MHz FREQ COUNTER,**  
**CAPACITANCE METER,**  
**INDUCTANCE METER,**  
**LOGIC PROBE**

**ENGINE**   **Standard Features plus -**  
**ANALYZER PLUS**   **TEMP. TACHOMETER &**  
 Model 95   **DWELL ANGLE TESTER,**  
 #990112   **DUTY CYCLE, 10M OHM**  
**\$19995**   **IMPEDANCE, ANALOG BAR**  
**GRAPH, K-TYPE TEMP**  
**PROBE, ALLIGATOR CLIP**  
**TEST LEADS, INDUCTIVE**  
**PICKUP CLIP, 6' TESTLEADS**  
**& DELUXE CARRYING CASE**

**Standard Features - Models 94 & 95**  
 ● DC/AC VOLTMETERS   ● AC/DC CURRENT  
 ● OHM METER   ● DATA HOLD   ● RELATIVE MODE  
 ● FREQ COUNTER to 4 MHz (Model 95)  
 ● AUDIBLE CONTINUITY TEST   ● DIODE TEST  
 ● MAX/MIN AVERAGE MEMORY RECORD  
 ● 10A HIGH-ENERGY FUSE PROTECTION  
 ● AUTO SLEEP & AUTO POWER OFF

KELVIN Electronics Serving You Since 1945

continues on a bit-by-bit basis. This effectively keeps the clock lines of all the active masters in sync. Since this method effectively maintains a synchronized clock, the actual arbitration for bus ownership takes place on the SDA line.

Two or more masters may generate a start condition at the same time. Arbitration takes place on the SDA line when SCL is high. What happens is that when a master transmits a high, while another master transmits a low, it discerns that the state of SDA has not changed and drops off the bus. That is, if the level of the data line doesn't match the level of the transmitted bit, the high man bows out.

This process can continue for a number of bits before the comparison fails. If both masters are trying to address the same device, the arbitration will continue into the actual data portion of the message. This method of using both the address and data for arbitration ensures that no loss of data will occur.

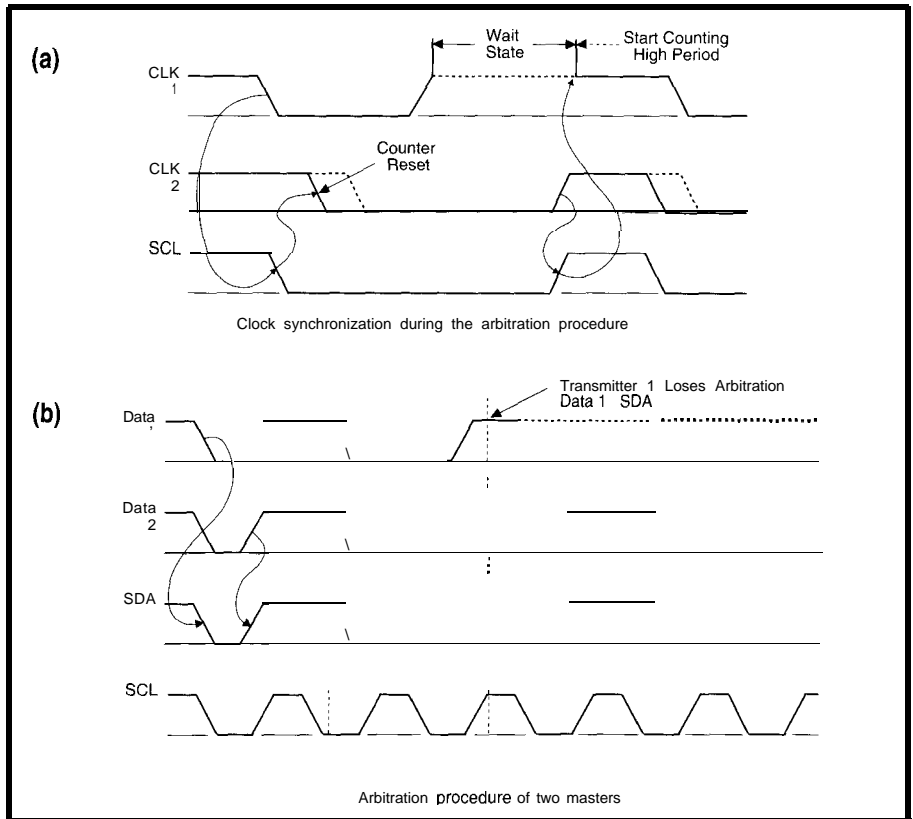


Figure 5-When there are multiple masters on the same bus, they must arbitrate for control. After synchronizing clocks (a), each begins transmitting while also monitoring the line (b). When one detects a difference, it backs off and either drops off the bus or becomes a slave receiver.

## MOVE OVER INTEL MICROMINT SOURCES 80C52 CMOS BASIC CHIP

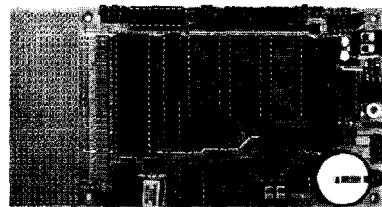
Micromint has a more efficient software-compatible successor to the power-hungry Intel 8052AH-BASIC chip. The 80C52-BASIC chip was designed for industrial use and operates beyond the limits of standard commercial-grade chips. Micromint's 80C52-BASIC chip is guaranteed to operate flawlessly at DC to 12 MHz over the entire industrial temperature range (-40°C to +85°C). Available in 40-pin DIP or PLCC

<b>80C52-BASIC chip</b>	<b>\$25.00</b>
OEM <b>100-Qty. Price</b>	<b>\$14.50</b>
<b>BASIC-52 Prog. manual</b>	<b>\$15.00</b>



**MICROMINT, INC.**  
4 PARK ST., VERNON, CT 06066  
TO ORDER CALL  
**1-800-635-3355**

## 8051 EMBEDDED CONTROLLERS With Lots of Extras!



We offer a full line of low cost 80C32 embedded controllers and software tools which are ideal for developing products, test fixtures and prototypes.

### Features Include:

- Low power CMOS design
- Up to 60K of code space and up to 60K of data space
- 5 to 15 volt operation
- Small form factor (3.5" \* 6.5") with prototyping area
- System diskette includes application notes
- Start at \$100

### Available Options:

- Multifunction Board adds A/D, 24 I/O lines and more!
- BASIC-52 or Monitor/Debugger in EPROM,
- C Compiler \$100 or BASIC Compiler for \$300

**Iota Systems, Inc.**  
POB 8987 • Incline Village, NV 89452  
PH: 702-831-6302 • FAX: 702 831-4629



**Listing I—**When you're using a processor that doesn't have a hardware PC interface, often the easiest way to add one is with software. While more processor intensive, it can be the cheapest solution

```

        public   Xmit_I2C_String
        public   Rec_I2C_String
        public   Xmit_I2C_Byte
        public   Rec_I2C_Byte

; Port I/O bits
SCL     equ     p0.0
SDA     equ     p0.1

; Macros...
MACRO   %Bit_Delay           ; Bit Delay
nop
nop
ENDMAC

MACRO   %Set_SCL             ; Set SCL
setb    SCL
jnb     SCL,$
%Bit_Delay
ENDMAC

MACRO   %Clr_SCL             ; Clear SCL
clr     SCL
%Bit_Delay
ENDMAC

MACRO   %Emit_Clock          ; Pulse SCL
%Set_SCL
%Clr_SCL
ENDMAC

MACRO   %Start               ; Start Sequence
setb    SDA
%Set_SCL

clr     SDA
%Bit_Delay
%Clr_SCL
ENDMAC

MACRO   %Stop                 ; Stop Sequence
clr     SDA
%Set_SCL
setb    SDA
%Bit_Delay
ENDMAC

RSEG    CODE

; Subroutines...
;
; Transmit a byte over the I2C bus
; input: acc contains byte to transmit
; output: cy = 0 if sequence completes
;         cy = 1 if unable to transmit

Xmit_Byte:
mov     r1,#8           ;8 bits to send
xbl:    rlc              a
        mov     SDA,c    ;put bit on pin
        %Emit_Clock     ;emit clock pulse
        djnz   r1,xbl    ;more bits to transmit?

; set up to accept ACK from slave device
setb    SDA             ;release SDA
%Set_SCL                 ;set SCL
jnb     SDA,xb2         ;jump if ACK seen
%Clr_SCL                 ;drop SCL
setb    C               ;set error code
ret

```

(continued)

A special case exists if a master also possesses the slave capability and loses arbitration during the address phase. Here, it's possible that the other master is trying to address it and the losing master must immediately switch over to its slave mode and become the receiver.

Figure 5 illustrates the arbitration procedure between two masters. As you can see, as soon as there is a difference between the transmitted data and the state of SDA, the losing master shuts off its data line and drops off the bus, or becomes the slave receiver depending on the situation.

## THE QUICK AND THE INTELLIGENT

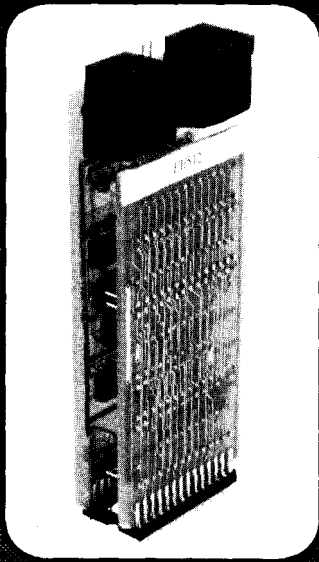
Microcontrollers with on-chip I<sup>2</sup>C hardware can be programmed to be interrupted in response to activity on the bus. This functionality can also be obtained using specifically designed I<sup>2</sup>C bus controller ICs that interface to standard microcontrollers. Of course, it can be taken for granted that I<sup>2</sup>C peripheral chips are always live (that is as far as continuously monitoring bus activity is concerned).

If you decide to interface your controller as a slave directly to the I<sup>2</sup>C bus using a strictly software means, the state of the bus must constantly be checked using software polling. Here, as is often the case, the intelligent operation of a microcontroller is no match for pure hardware. This bus polling activity can consume a lot of time that could be better spent doing more useful functions. The problem can be alleviated, to a great extent, by using a special start procedure that is much longer than normal.

The special start procedure is accomplished by sending a standard start sequence followed by a special start byte which consists of the binary pattern 00000001. In this case, the receiving controller can sample SDA at a low rate until one of the zeros is detected. Once the low level is sensed, the controller can switch to a higher sample rate to find the repeated start condition before normal communications commences.

This sequence causes no problems for hardware-based receivers since they

# EPROM EMULATORS



# \$129

AND UP

- Downloads through standard PC compatible printer port
- Includes RJ-11/12 adaptor and 7' modular cable
- Generates RESET and /RESET
- Accepts Binary, Intel HEX, Intel Extended HEX and 'S' files
- Unique Vertical design minimizes mechanical interference and eliminates the possibility of noise cross-talk or transmission-line effects from the use of a cable
- Multiple devices (except EE256) can be daisy-chained for 16/32 bit or multi-bank target systems

EE256 (2764-27256) \$129  
 EE512 (2764-27512) \$159  
 EE1M (2764-27010) \$199  
 EE2M (2764-27020) \$299

=>Call for EE2M availability<=<

10 day MONEY-BACK GUARANTEE  
 90 day repair/replace warranty

## Technical Solutions

PO BOX 462101  
 Garland, TX 75046-2101

Call or FAX for ENGINEERING SPECS.  
 Voice or FAX: (214) 272-9392

### Listing 1-continued

```

xb2:      %Clr_SCL                ;drop SCL
          clr                    c                ;set complete code
          ret

; Receive a byte over the I2C bus
;   output:  acc contains received byte
           cy is dummed up with a 0

Rec_Byte:
          niob                    r1,#8          ;8 bits to receive
          %Set_SCL                ;set SCL
          mov                      c,SDA        ;pick up data bit
          rlc                      a
          %Clr_SCL                drop SCL
          djnz                     r1,rbl       more bits to receive?
          clr                      c            must complete ok!
          ret

; Public routines...

; Transmit address and data bytes over I2C bus
;   input:  acc contains slave address
           b contains register address
           dptr points to source buffer
           r0 contains byte count
;   output: cy = 0 if sequence completes
           cy = 1 if unable to transmit

xd_fault: ;bus fault
          setb                      c            set error code
          ret

Xmit_I2C_String:
          jnb                      SCL,xd_fault ;jump if bus fault
          jnb                      SDA,xd_fault ;set start condition
          call                     Xmit_Byte    send slave address
          jc                      xd2          jump on error
          mov                      a,b
          call                     Xmit_Byte    send register address
          jc                      xd2          jump on error
          movx                     a,@dptr     pick up data byte
          inc                      dptr
          call                     Xmit_Byte    send data byte
          jc                      xd2          jump on error
          djnz                     r0,xd1      more bytes to send?

; set stop condition, return code is already in cy
xd2:      %Stop                    set stop condition
          ret

; Transmit address and receive data bytes over I2C bus
;   input:  acc contains slave address
           b contains register address
           dptr points to destination buffer
           r0 contains number of bytes to receive
;   output: cy = 0 if sequence completes
           cy = 1 if unable to receive

rd_fault: ;bus fault
          setb                      c            ;set error code
          ret

Rec_I2C_String:
          jnb                      SCL,rd_fault ;jump if bus fault
          jnb                      SDA,rd_fault

          %Start                    ;set start condition
          push                     acc
          call                     Xmit_Byte    ;send slave address
          pop                      acc
          jc                      rd3          ;jump on error
          xch                      a,b
          call                     Xmit_Byte    ;send register address
    
```

(continued)

## Listing I-continued

```

        jc          rd3          ;jump on error
        %Start      ;set repeated start
        mov         a,b
        setb        acc.0       ;indicate read operation
        call        Xmit_Byte   ;send slave address again
        jc          rd3          ;jump on error
rdl:    call        Rec_Byte    ;receive data byte
        movx       @dptr,a     ;store data byte
        inc        dptr

; Send ACK on all but last byte
        djnz       r0,rd4      ;jump if not last byte
; sequence complete, return code is already in cy
rd2:    setb        SDA        ;set SDA (no ACK)
        %Emit_Clock ;emit clock pulse
rd3:    %Stop      ;set stop condition
        ret

; send ACK and continue receiving
rd4:    clr         SDA        ;drop the Ack bit
        %Emit_Clock ;emit clock pulse
        setb        SDA
        sjmp       rdl        ;keep receiving

; Transmit address and a data byte over I2C bus
; input: acc contains slave address
;        b contains data byte
; output: cy = 0 if sequence completes
;         cy = 1 if unable to transmit
xdb_fault: ;bus fault
        setb        c         ;set error code
        ret
Xmit_I2C_Byte:
        jnb        SCL,xdb_fault
        jnb        SDA,xdb_fault ;jump if bus fault
        %Start      ;set start cond.
        call        Xmit_Byte   ;send slave address
        jc          xdbl       ;jump on error
        mov         a,b
        call        Xmit_Byte   ;send data byte
; set stop condition, return code is already in cy
xdbl:   %Stop      ;set stop condition
        ret

; Transmit address and receive a data byte over I2C bus
; input: acc contains slave address
; output: b contains received byte
;         cy = 0 if sequence completes
;         cy = 1 if unable to receive
rdb_fault: ;bus fault
        setb        c         ;set error code
        ret
Rec_I2C_Byte:
        jnb        SCL,rdb_fault
        jnb        SDA,rdb_fault ;jump if bus fault

        %Star      ;set start condition
        setb        acc.0     ;indicate read operation
        call        Xmit_Byte   ;send slave address
        jc          rdbl       ;jump on error
        call        Rec_Byte    ;receive data byte
        movx       b,a        ;store data byte
; sequence complete, return code is already in cy
        setb        SDA        ;set SDA (no ACK)
        %Emit_Clock ;emit clock pulse
rdbl:   %Stop      ;set stop condition
        ret

```

will reset on the repeated start condition and will ignore the start byte. Note that an acknowledge-related clock pulse is generated after the start byte. This is a dummy clock that exists only to conform to the standard data format on the bus and no device should attempt to assert an acknowledgment.

Although I<sup>2</sup>C defines even more capabilities, I'll not belabor this any further. I'm sure by now you can appreciate the capabilities that have been imparted to these two wires by the designers.

## SOFT I<sup>2</sup>C

So, now you have a serial bus architecture that supports multimaster communications, a bunch of special features, and looks like it's set up to handle just about any eventuality. As I mentioned, you can get microcontrollers with built-in I<sup>2</sup>C hardware support. If none of these suit your taste, you can get dedicated I<sup>2</sup>C controllers that can be lashed to just about any processor you like. With all this available hardware you can easily tap into the I<sup>2</sup>C feature set. What do you do next!

Well, in spite of the capabilities of the I<sup>2</sup>C bus, a lot of people will be content to just run the whole thing under control of a software loop. The fact that this is feasible, confirms the architecture's flexibility. Taking this approach, you would most likely only use a fraction of the capabilities of the bus. But a little may be all you need. While this can be a restrictive approach, as far as CPU utilization is concerned, it opens up a lot of peripheral functions at the cost of just two pins.

The software-based I<sup>2</sup>C driver loop I'm presenting here is designed to interface to block-oriented devices—that is, devices that have addressable registers. I'm sure you can see that adapting these to handle simple peripherals is...well, simple. Basic block read and write operations are supported with slave addressing and register addressing support built-in. Let me begin by briefly describing the code starting from the smallest component parts.

The macros appearing at the start of the listing should be pretty self explanatory. My motive in coding the fundamental functions as macros was primarily driven by a desire to keep the source code as simple and uncluttered as possible. I found out that the software implementation of the I<sup>2</sup>C protocol was not quite as clean as I would have liked, so I tried to encapsulate the ugliness as much as possible.

The subroutines that follow should also be clear in their intent. They simply handle the signaling on the pins to effect the data transfers; transmitting and receiving bytes. Here, the embedded macros hopefully make the code a bit more readable. You will notice that I use the carry bit as a success/fail indicator. This indicator is conveyed up to the application layer. Status information is useful in identifying bus faults and malfunctioning peripherals. Indeed, this is an old trick whereby you can defer from directly handling the possible error conditions. Let the more intelligent code go at it! Divide and conquer!

Finally, the public transmit and receive routines handle the details of framing the transactions with the start and stop sequences and manage the peripheral's addressing requirements. The register selection procedure for the write function is handled by first sending the obligatory slave address to select the chip and then writing the register address. Following this, the data is immediately written to the slave. Reading requires the same initial sequence, but following this, a repeated start condition must be asserted. The slave address is again presented, but now a read operation is indicated (LSB=0). Following this, data can be streamed from the peripheral. In both cases the byte count for the operation is held in `r0; d p t r` is the source or destination pointer as appropriate. Note that the routines allow writing or reading multiple bytes. This is possible since most registered peripherals have an auto-increment capability that bumps up the internal register address generator at the conclusion of each write or read.

## WRAPPING UP

It's always a good idea to cover some theory when exploring a new topic, but it's equally important to apply it. Right now I'm rounding up a bunch of I<sup>2</sup>C peripherals and controllers. Next time I'll have some of this stuff wired up. I'm hopeful I can get it to do something useful. □

*John Dybowski is an engineer involved in the design and manufacture of hardware and software for industrial data collection and communications equipment.*

## SOURCE

Signetics Company  
811 E. Arques Ave.  
P.O. Box 3409  
Sunnyvale, CA 94088-3409  
(408) 991-2000

## IRS

419 Very Useful  
420 Moderately Useful  
421 Not Useful

## ► Project Parts ◀

Firmware Flyers (prices shown are postpaid in US/ with any parts order)	
8051 Firmware Debugging Techniques (65 pages w/ 3.5" diskette)	\$18 /15
Introduction to the 8051 Instruction Set (46 pages)	\$10 /8
8051 Instruction Reference Card	\$3 /2
8255 Cheat Sheet Reference Card	\$3 /2
Embedded '386SX Project Parts (INK 33, 34, 35)	45.70
LD273 dual IR LED (bright, wide beam)	2.10
UGN3503U Ratiometric linear Hall Effect sensor	2.25
IR3C02A laser diode controller (±5, for LT022MC/LN9705P laser)	2.30
IR3C07 laser diode controller (+5V, for LT022PD/LN9705 laser)	2.85
PH302 fast IR photodiode	3.10
GP1U52Y 40 kHz IR receiver (side-looking)	4.00
ULN3751Z Power op amp (±3V to ±13V supply, 3.5 A output)	4.60
IS1U60 38 kHz IR receiver	4.80
UDN2993B Dual full H-bridge bipolar stepper motor driver	5.45
IRSAMPLE parts (PH302, LM311, etc. See MCIR-Link article, INK 29)	5.70
Excellent IR filter (opaque to visible light, 35 mm slide mount)	6.75
MC145030 IR encoder/decoder	6.75
DS1232 Micromonitor watchdog	6.75
UCN5841A Serial-input, 8 latched 500 mA sink drivers	6.90
UCN5895A Serial-input, 8 latched 250 mA source drivers	7.50
UCN5804B Unipolar stepper motor translator/driver	8.00
CS212 S-ART I/O network/security monitor	8.10
DS2400 Silicon Serial Number (2 chips)	8.40
DS1210 NVRAM power controller	9.75
DS1231 Power monitor	9.90
DS1202 Serial clock/calendar	10.20
IR I/O: IS1U60, LD273, CD4047, 2N2907, red LED, schematic (IR-Link)	10.40
MT8809 8x8 analog crosspoint	11.50
MAX691 Power supervisor	11.85
75T204 DTMF decoder	12.00
IL300 linear optoisolator	12.70
MAX233 self-contained +5V powered RS-232 interface	14.25
MT8880 DTMF encoder/decoder (bus interface)	18.90
PL513 Send-only X10 power line interface	20.00
MAX134 digital millimeter	33.00
MAX252 optically isolated +5V powered RS-232 interface	64.00

UPS Ground/2nd day/Next day \$6.50/18 to 48 US states, COD add \$4.50, Canada \$6 via USPS Air Small Packet, no CODs. Check, MO, or COD only, no credit cards. POs add \$50, but call first. CT residents add 6% sales tax. Quantity discounts start at five parts. Data Sheets included.

Call/write/fax for seriously tempting catalog...

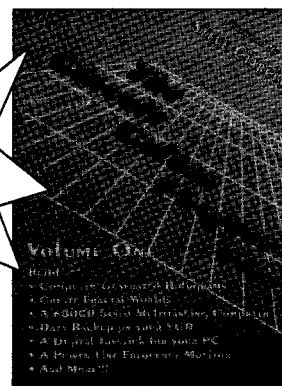
### Pure Unobtainium

► Your unusual parts source ◀

89 Burbank Road • Tolland, CT 06084-2416  
FAX/voice (203) 870-9304

## The Circuit Cellar Project File, Volume 1

**VOL. 2  
COMING  
THIS  
SUMMER!**



The Circuit Cellar Project File, Volume 1 has over 200 pages of new and expanded hands-on projects and tutorials. The Computer Applications Journal's editors have chosen a dozen of the top projects from the Circuit Cellar Design Contest, independent submissions, and top-response articles to make a book with something for every interest!

now only **\$17.95!** (includes domestic delivery\*)

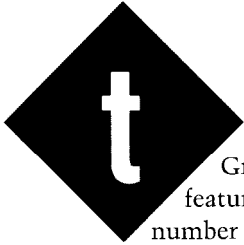
Order your copy today!

\*\$17 95 Visa, Mastercard, Check, or Money Order (US Funds Only)

(Add \$200 for delivery to Canada or Mexico, \$4.00 for delivery to other non-U.S. addresses)

### The Circuit Cellar Project File, Vol. 1

4 Park Street, Vernon, CT 06066 • Tel: (203) 875-2 199 • Fax: (203) 872-2204



his month's collection of patents on Graphics and Video support this issue's feature-topic theme by presenting a number of current developments in this

rapidly evolving area of technology. These abstracts range from a couple of fundamental patents in multimedia and virtual reality to design techniques for enhancing, speeding up, and compressing video and transmission of video information in television format.

Two of the most talked about (and esoteric) areas in computer video these days are multimedia and virtual reality. The first two patents demonstrate moves to get early footholds in these emerging areas. Abstract 1 presents a multimedia interface approach that very simply and flexibly piggybacks multimedia onto existing text-only applications. The idea is to have the software send commands to multimedia playback units in parallel with the textual data. In this manner, the current text-only application is augmented with additional speech and video images. I would suspect that the full power of such a scheme would

be realized only if it were to become a standard that allows a wide variety of multimedia devices and software applications to coexist harmoniously.

Abstract 2 provides a similar "fundamental" capability, but in the field often known as virtual reality. For a wide range of applications, and specifically in military displays, it is necessary to superimpose images of real-life scenes and computer-generated objects. For the resulting image to look realistic, it is crucial that three-dimensional placement and perspective between the two video sources be matched. While the abstract does not present the actual techniques used, it begs the reader to review the entire patent to see what novel approaches might be used. I've discussed in past columns how to get complete patents.

The next three patents relate to generation, control, and manipulation of computer graphics images. The first, by Philips Corporation, presents a hardware approach to the hidden-surface removal problem which would offer super-fast performance. As Abstract 3 explains, normal x, y, and color data are augmented by storage of z (depth) information in video memory. Essentially, a special-purpose graphics processor-most likely incorporated within a graphics controller chip design-uses these data types to determine whether or not new data should be written to video RAM.

Patent Number	4,931,950	
Issue Date	1990 06 05	
Inventor(s)	Isle, Brian A.; Bloom, Charles P.; Butler, Arch W.; Spoor, David; Wunderlin, David J.; Bedros, Renee	
State/Country Assignee	MN Electric Power Research Institute	
US References	4,578,555 4,602,134 4,617,661 4,644,478 4,644,479 4,649,515 4,654,852 4,670,848 4,757,463 4,776,016 4,816,208	
US Class	3641513 364/188364/551.OI 3641579 3641514 3641200 3641274.7 3641275.7 381/1 10	
Int. Class	G06F 15/46 G06F 1 1/30	
Title	Multimedia interface and method for computer system	
Abstract	A multimedia interface presents information and receives user commands for a computer system. The multimedia interface operates in parallel with another application software module, such as an expert system. To add multimedia features to the application software module, the module is modified so as to generate multimedia commands at the same time as it displays text on a text monitor. The multimedia commands, which are held in a queue, provide additional information in the form of video images and generated speech corresponding to the displayed text. In addition, the multimedia commands are split into at least two sets: one set which is dispatched to the user substantially immediately after displaying the corresponding text, and one set which is dispatched only upon request by the user. In the preferred embodiment, the multimedia interface presents information to the user through text, graphics, video, speech production, and printed output. User inputs are made through a special- function keypad and voice recognition. The preferred embodiment is a portable expert system which fits in a single portable suitcase-sized package.	

# PATENT TALK

2

Patent Number 4,970,666  
Issue Date 1990 11 13

Inventor(s) Welsh, William T.; Pittman, Kenneth B  
State/Country NC  
Assignee Land Development Laboratory, Inc.

US References 4,425,581 4,600,200 4,645,459 4,684,990 4,736,306 4,805,121 4,835,532

US Class 3641522 3641521  
Int. Class G06F 15/72

Title Computerized video imaging system for creating a realistic depiction of a simulated object in an actual environment

Abstract A system and method for producing highly realistic video images which depict the appearance of a simulated structure in an actual environment, and provides for accurate placement of the structure in the environment and matching of the perspective of the structure with that of the environment so that a highly realistic result is achieved. The system includes a video input means, such as a video camera and video recorder, by which a video image of the actual environment may be captured. A graphics processor unit receives the video image from the video input means and stores it in rasterized form. Field data input means is provided for receiving field location data regarding the precise location of the captured video image and field perspective data regarding the perspective of the captured image. Object data input means is also provided for receiving data, such as CAD data for example, for a three-dimensional model of a simulated object which is proposed to be included in the environment. From the three-dimensional model of the object, a two-dimensional perspective representation of the object is generated, which accurately matches the perspective of the captured video image. The thus generated two-dimensional perspective representation is then merged with the rasterized video image and accurately positioned at its proper location in the environment.

3

Patent Number 4,924,415  
Issue Date 1990 05 08

Inventor(s) Winsler, Paul A.  
State/Country GBX  
Assignee US. Philips Corporation

US References 4,609,993 4,625,289 4,748,572

US Class 3641522 3641521 3641518 3401750  
Int. Class G06F 15/68

Title Apparatus for modifying data stored in a random access memory

Abstract Input data defines the address (X,Y), and an input color (RGB) and depth (Z) for a picture element (pixel) within a stored image. In order to perform hidden-surface removal (HSR), current depth values are stored for each pixel and compared with the input depth value to determine whether or not input data should be written to define a new color and depth for the pixel at (X,Y). The color and depth values are stored in a color RAM (9) and z-RAM (64). To obtain a speed advantage when modifying a series of consecutive pixels and one row of the RAMs, the current depth values are read and compared in advance for each pixel, during the writing period of one or more preceding pixels. The apparatus comprises a control and arithmetic unit (42), and an HSR control circuit (60) in addition to the color RAM (9) and z-RAM (64). The apparatus uses readily available video DRAM chips to provide a z-RAM with two data ports (62,66). The apparatus may form part of an electronic graphics system.

# PATENT TALK

4

Patent Number	5,025,249
Issue Date	1991 06 18
Image Disc #	This patent is on PatentImages disc# 1991\045
Inventor(s)	Seiler, Larry D.; Pappas, James L.; Rose, Robert C.
State/Country Assignee	MA Digital Equipment Corporation
US References	4,204,206 4,386,410 4,412,294 4,484,187 4,354,376 4,545,070 4,550,315 4,642,790 4,694,288 4,700,320 4,710,761 4,710,767 4,727,425 4,752,893 4,772,881
US Class Int. Class	3401721 3401723 3401729 3401734 40/522 3641522 G09G 1/00
Title	Pixel lookup in multiple variably sized hardware virtual color maps in a computer video graphics system
Abstract	This invention adds a window-dependent base value to the pixel values read from a frame buffer or other source of pixel values. The base value points to the base of the color map for that window, which is allocated within a larger, physical color map. Each window can access physical color map entries starting at its base value and extending up to the base value plus the maximum pixel value used in that window. Adding a window-dependent base value to the pixel values for each window allows different windows to use different color maps, each of which can be allocated to any contiguous set of entries in the physical color map. Each window's virtual color map need only use as many entries in the physical color map as there are entries in the virtual color map. Finally, virtual color maps can be compacted or otherwise reallocated in the physical color map without requiring changes in the pixel values stored in the frame buffer. Only the color map base values stored for each window need be changed.

Pixels representing points physically behind other pixels (at a greater z-value) would not be written.

The DEC patent shown in Abstract 4 provides a hardware means for achieving enhanced and potentially more dynamic and interesting displays within a windows environment. Normally, the entire display screen is limited to a given number of colors (often 4, 16, or 256—corresponding to 2, 4, or 8 bits per pixel) even though these pixels may be chosen from a larger color palette. This

invention permits each window to use its own set of colors chosen from the large palette. In this manner, for example, the compact memory utilization of four-bit color mode might be retained while achieving a much more interesting display in which a different set of 16 colors is used within each window. Of course, software changes would be required to store the current window number along with each pixel. This would offset much of the memory savings, however it still offers the prospect for strikingly more

5

Patent Number	5,033,105
Issue Date	1991 07 16
Image Disc #	This patent is on PatentImages disc# 1991\053
Inventor(s) State/Country Assignee	Atkinson, William CA Apple Computer
US References	3,882,454 4,021,778 4,441,205 4,622,545 4,631,521 4,646,356 4,729,127 4,736,437 4,743,973 4,837,848
US Class Int. Class	382/56 358/261.1382/48 382122 G06K 9/36
Title	Video compression algorithm
Abstract	A method for encoding compressed graphics video information and decoding such information. The method consists of enriching the video information in zeros through shifting and Exclusive ORing video with itself. A number of methods are attempted in the shifting and Exclusive ORing process in order to determine the method which yields the optimum zero-enriched image. The zero-enriched image is then encoded and the encoded information stored. Upon retrieval, the information is decoded and an Exclusive OR and shifting process is done to obtain the original video information.

# PATENT TALK



Patent Number	4,941,040
Issue Date	1990 07 10
inventor(s)	Pocock, Terrence H.; Coumans, Peter J.; McNorgan, Richard M.; Hart, George M.
State/Country Assignee	CAX Cableshare, Inc.
US References	4,450,477 4,450,481 4,518,989 4,538,176 4,734,764 4,757,371 4,761,684 4,780,757 4,780,758 4,789,895 4,792,849
US Class Int. Class	358/86 455/4 H04N7/10
Title	Cable television system selectively distributing prerecorded video and audio messages
Abstract	<p>A method of, and a system for, selectively delivering still television video with accompanying audio to home subscribers over a cable television system for advertising, promotional, or educational purposes. A maximum number of home subscribers can interactively request presentations of their own choosing to be displayed on their home television sets. Only one standard television channel is required for transmission of still video with accompanying audio to serve 300 concurrent users. No equipment is required in the subscriber's home. The video is presented as still frames from one of a number of videodisc players, transmitted over one television channel during the appropriate time interval of 1/30th (or 1/25th) of a second. Such video frames, which may also contain overlaid graphics information, are uniquely addressed to a remote storage device. Unused bandwidth is used for the transmission of up to 300 discrete audio messages. The remote storage device identifies the appropriate video still frame, stores it, combines it with the corresponding audio message and conveys both to the home subscriber's television on a preselected channel. By uniquely addressing video frames to the remote storage device, either 30 (or 25) different video frames per second can be conveyed on one television channel to 30 (or 25) different remote storage devices for retransmission to home subscribers. Thus, if a home subscriber sees a given still video frame for 10 seconds, his remote storage device need not be updated for those 10 seconds, enabling the system to transmit 300 (10 seconds.times.30/second) different video frames to 300 other remote storage devices, thereby serving 300 concurrent users.</p>

dramatic and useful displays, particularly with 256-color windows.

Apple Computer presents an intriguing patent in Abstract 5 which presents a novel means of compressing video data. The raw data is first massaged by an algorithm that essentially acts like a pseudorandom noise function (like a feedback shift register) in order to introduce more zeros into the data. Apparently, a somewhat hit-or-miss approach is used to find a "good" encoding key by which the data may be compressed to a greater degree than without this manipulation. While only the algorithm is mentioned in the abstract, there would seem to be application areas in which a hardware-implemented and pipelined "zero enricher" followed by a conventional (run length or other) encoder could rapidly compress video data, perhaps for transmission as well as storage.

The next two patents relate to video information in TV form. Abstract 6 presents a method for supplying any one of many still images (and accompanying audio) to a home TV, for example for advertising purposes via cable. The scheme sends each "still" in a single TV frame that is stored at the destination in a frame buffer, from which it is played out to the TV receiver. The number of still images that may be carried over a single TV channel depends on how rapidly the viewer requires new visual information. Using their

example, if the viewer is content with viewing one image for 10 seconds, then (since a frame represents 1/30 second) the single channel can carry 300 still images in this ten-second period. Note that if the requirement permits the image to be updated only once a minute, then this scheme could handle 1800 images per channel. Although not mentioned in the abstract, this scheme could have applicability beyond cable TV. For example, it could be used to broadcast images within a department store or a factory. It could also support nonvideo information in essentially the same "frame-multiplexing" manner, because the frame buffer really doesn't care what meaning the information has.

I was a bit surprised to find a patent issued to The Weather Channel, for we tend to think first of the weather forecaster on our screen describing tomorrow's showers rather than the technology behind the scene. The scheme shown in Abstract 7 presents a method for centralized distribution of weather information (video, audio, and data) to remote receivers in a TV network. The novel scheme not only permits remote receivers to capture just the information that is pertinent to each, but also for the central computer to send command sequence "scripts" to the receivers to tell them, in effect, what to do. In a sense, what we have here is a multimedia distribution network.



# PATENT TALK

7

Patent Number 5140,419  
Issue Date 199208 18

Inventor(s) Galumbeck, Alan D.; MacKinnon, Russell D.; Pincock, Douglas G.; Reid, Frederick A.

State/Country GA  
Assignee The Weather Channel, Inc.

US References 4,131,881 4,383,257 4,394,687 4,429,385 4,430,731 4,528,589 4,725,886 4,916,539

US Class 358/142 3401825.44  
Int. Class H04N 71087

Title Communications system

Abstract A communications system having centralized management and multiply hierarchical addressing schemes is disclosed. The system may be used in connection with supplying video, audio, and data such as weather-related text, graphics, and information to affiliated receivers in a network for broadcast or display. Receivers may be addressed singly or in groups and allowed to determine their own addresses from information keyed, directly or indirectly, to a receiver characteristic such as the unit serial number. Lists of commands denominated "scripts" and transmitted to receivers are used for controlling the various modes or states of the receivers.

Finally, with video information so pervasive, there always seems to be the need or desire to restrict someone's access to it. Abstract 8 purports to be a novel means of controlling children's viewing of TV through the use of a microprocessor-based, parent-programmable "video control system." Even with its LCD readout, I wonder how many parents will go through the trouble of programming it (based on all the complaints regarding the difficulty of programming VCRs), or how the system knows which child (and his friends and siblings!) is watching at any given moment. But wherever a problem is perceived, there is always someone ready to patent a solution to it! ☹

*Russ Reiss holds a Ph.D. in EE/CS and has been active in electronics for over 25 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 new products. He may be reached on the Circuit Cellar BBS or on CompuServe as 70054,1663.*

Patent abstracts appearing in this column are from the Automated Patent Searching. (APS) database from:

MicroPatent  
25 Science Park  
New Haven, CT 065 11  
(203) 7865500 or (800) 648-6787

## IRS

422 Very Useful    423 Moderately Useful    424 Not Useful

8

Patent Number 5,168,372  
Issue Date 1992 1201

Inventor(s) Sweetser, David J.  
State/Country c o

US References 3,581,029 3,833,779 3,879,332 4,145,617 4,246,495 4,317,213 4,348,696 4,358,672 4,510,623 4,566,033  
4,588,901 4,718,107 4,956,825 5,051,837 5,053,884 5,060,079

Title Video control system

Abstract A video viewing control system which permits the parent to enter a viewing "allowance"(time) for each child and which disables viewing of a television by disrupting the television's Radio Frequency (RF) input signal or video input signal when the child has watched television for his allowable time. The system includes a single chip microprocessor and a liquid crystal display which permits display of graphical symbols to the child for easy comprehension. The system is battery-powered to protect against power outages. The child's viewing time can be set on a daily basis (potentially different for each day of the week) or can be done on a weekly basis (a lump-sum covering the entire week). Block out times (during which television viewing is disabled) can be programmed for each child on the hour and half-hour for any day of the week. The system calculates and displays each child's average viewing time per day and the total viewing time over any desired period.

The Circuit Cellar BBS  
300/1200/2400/9600/14.4k bps  
24 hours/7 days a week  
(203) 871-1988—Four incoming lines  
Vernon, Connecticut

*A few months back I wrote about our new high-speed modems. We've just finished completely replacing the rest of the hardware on which the Circuit Cellar BBS runs. The combination of the high-speed modems, the QWK support, and increased usage was really taxing the older AT-style hardware we had been using. We're now running on a 33-MHz '386SX with 280 megabytes of disk space. If you gave up calling because the system response had gone down or were holding off on that large upload, give it another shot.*

The first two threads this month are follow-ups to last issue's *ConnectTime*. In that issue, Tom Maier asked about the use of a series resistor with a crystal in a processor oscillator circuit (specifically, with the Microchip PIC). In the first thread this month, a reader offers another explanation for that series resistor.

In the second thread, another robotics enthusiast jumps into the discussion started last month about robotics' past and future. This topic continues to be hot on the BBS, so if you want to join in, it's not too late to call.

Next, we have some trials, tribulations, and tips from someone who's been through FCC (and other agency) testing and lived to tell about it.

Finally, we look at what's involved in frying to make a toy car follow a buried guide wire.

## Crystals and series resistors

**Msg#:11141**

**From: BILL HAWKINS To: TOM MAIER**

Tom, I saw your question about the series resistor in *ConnectTime*. I have just had occasion to "discover" the reason for series resistors with crystals; up to now I had not used them either. We recently took an embedded system through FCC approval. It had only one oscillator circuit, using a 24-MHz crystal. We had wild emissions at 120.00 MHz, which sure surprised us. The guy running the test thought that this noise was the fundamental frequency, since there were no emissions at 60 MHz.

After a long weekend worrying about the problem, I remembered something about square waves and odd harmonics. Sure enough, 120 is a 5th harmonic of 24. The problem was the square wave on the crystal. The atmospheric coupling of the circuit peaked around 120 MHz, thus our radiated noise. You can't get rid of the harmonics and still have a square wave, so the only thing you can do is

lower the amplitude. This is where the series resistor comes in. Instead of banging a square wave at 5 V, the resistor lowers it to somewhere around 2 V (in our case.) Of course, this means the crystal is no longer a digital signal. The drive circuit actually runs in linear mode. You still have to hit the proper thresholds to make the circuit switch though.

We used a 1k series resistor, and this put our amplitude in the mud. This also prevented the circuit from oscillating. The fix for that is a parallel resistor across the crystal (470k in our case). This resistor biases the circuit to ensure oscillation. Higher values of series resistor result in lower-amplitude oscillations. The parallel resistor, carried to 0 ohms, would oscillate at the propagation frequency of the crystal driver, so it must be a fairly high value to keep from overwhelming the crystal. It also must be low enough to help the crystal swing over the input threshold. You aim at an amplitude high enough to prevent stray noise from affecting oscillation, but low enough to prevent radiating odd harmonics to the world. Hope this helps..

**Msg#:11152**

**From: TOM MAIER To: BILL HAWKINS**

Well, I hadn't thought about radiated noise as the reason for the series R. The series R would decrease the harmonics on the oscillator. What is confusing to me is the Microchip data sheets say the crystal manufacturer would recommend the value of the series R. This seems silly to me. How would they know? It would be a function of the oscillator, and Microchip made the oscillator.

Increasing the series R would cause more rounding of the edges of the square wave, thus reducing the amplitude of the 5th harmonic and the emitted signal. This is due to RC filtering of the feedback of the oscillator.

I have the feeling this has to do with the power handling capability of the oscillator in the PIC. If the crystal feedback is too strong, then it blows the oscillator.

You are right about the harmonics, but I think there is some other answer needed here concerning the power handling capability of the oscillator and/or the crystal.

**Msg#:11376**

**From: BILL HAWKINS To: TOM MAIER**

I am using a Xilinx part, and they say the same thing.. The series R may be needed, consult crystal manufacturer..

# CONNECT TIME

I did talk to our crystal FAE, he didn't know anything about it. Then he checked some of his modem applications, and found the series R and the explanation that it was for FCC compliance. I have never heard of overloading an oscillator circuit with a crystal, but then I suppose anything's possible. I only thought it may be the same problem because both manufacturers used the same verbiage. I think they send you to the crystal manufacturer only because they don't want to have to deal with it. It really is more a system problem than a crystal or oscillator constraint. I think the proper R is much more dependent on board layout and choice of damping capacitors.

---

## Robotics musings

Msg#:11137

From: ROBERT NANSEL To: WALTER BANKS

Asking "What happened to robotics?" is like asking "Whatever happened to kids these days!" In other words, the question is too general; robotics isn't a monolithic industry (never was). You have to separate the maturing (i.e., boring) industrial-style manipulator market from the emerging (i.e., exciting!) mobile service robot market. They are very different animals.

First, mobile service robots are a harder sell. It isn't easy finding truly economic uses for mobile robots. Right now the field is pretty much limited to office mail delivery, hospital meal delivery, warehouse security, nuke plant maintenance, and bomb disposal. The common thread among all of these is they are all applications where cost isn't the overriding consideration and the robot's domain is controlled and limited.

For example, TSR, a company in Connecticut, makes a machine they call the HelpMate for hospital meal delivery. The HelpMate is about the size of a washer or dryer, weighs more, and is one dazzling bundle of technology. When Jake Mendelssohn and I visited TSR's facility in Danbury last September, I was fabulously impressed with the machine. Here they've really really solved the tough problems of autonomous navigation.

The only concession they make in the environment for the robot is to put a foot-long piece of 3M reflective tape on the ceiling at one location in their office suite. The HelpMate uses that as a benchmark to fix its position at the start of its rounds. It uses no buried wires to guide it, no UV bar codes on the ceiling, no IR beacons to mark doorways. The HelpMate instead relies on an internal CAD model of the floor plan, a host of ultrasonic and IR obstacle detectors, dead reckoning, and the deft use of structured light projection to reduce the data it must crunch from its CCD

camera. It's designed to navigate a known environment consisting of corridors, elevators, and doorways.

The various subsystems of the HelpMate are controlled by satellite 68HC11 processors networked by an RS-485 multidrop scheme. A 68k waves the baton for the 'HC11 chips. Interestingly, they chose a '386 motherboard to do the vision processing from the CCD IR camera. The '386 is able to handle this task because they use structured light (two xenon strobes with focusing optics to create horizontal fans of IR filtered light). One projector is mounted low, about 12 inches off the floor, the other a bit higher at about 18 inches. They pulse the strobes alternately about five times a second.

The IR camera, mounted inside the robot above the strobe projectors, looks forward and down at about a 45° angle. Its field of view covers a patch of floor between three and fifteen feet in front of the robot. What the IR camera sees when the robot is in a corridor is a binary image with two sets of converging IR stripes (the walls of the corridor) alternating between 12 and 18 inches from the floor. If there is no obstacle in front of the robot, that's all it sees. If there is an obstacle at least 12" tall, the robot will see a contour that disrupts the straight light stripes. If a doorway is present, the robot will see that as a discontinuity in one of the straight light stripes. Using this method, they so reduce the amount of data that needs to be passed off to the master 68k processor, that the '386 is interfaced to the multidrop network through an 'HC11. The robot can make free-space calculations (that is, determine where there is room enough for the robot to pass) in something like 20 milliseconds. That's right-it navigates in real time.

Anyway, I mention all of the above to drive home a point. The HelpMate is state of the art. I know of no other system that can touch it. It costs something like \$60k if you wanted to buy one. Mostly TSR leases them to hospitals for the strategic amount of \$6/hr., 24 hours a day, 365 days a year. In-the-field experience in real hospitals shows over 98% successful trip completion. For their market, the HelpMate will ultimately be successful because it makes economic sense. But..

\$60k is a lot of money. There are few groups and even fewer individuals who can contemplate spending that much for production models [not to mention R&D] without an extremely clear idea what the market is.

So, mobile service robotics is in the classic chicken-and-egg problem of there being no money to develop mobile robot technology until a market is established, but a market can't be established until there are robots developed that do something that folks need doing, and developers don't know what that killer application might be, so round you go.

The thing freezing out most tinkerers from just blindly whipping together a robot and throwing it on the market to

# CONNECTIME

see what happens is cost. If we can find an application that inherently requires low-cost hardware, where the entry costs can be something more reasonable than \$60k (say under \$1000, but preferably under \$500), then we'll see a level of experimentation and development that will allow us to eventually stumble onto the VisiCalc or PageMaker of mobile robotics. What would that first low-cost application be? My guess is it will be some kind of robotic sport/hobby. Consider the situation with R/C model cars, boats, and planes. Here is a market with millions of units moved each year. The average R/C modeler sinks, say, \$400 into one model, probably owns more than one model, and will probably buy another model within the coming year. If you doubt the viability of this marketplace, thumb through an R/C modeling magazine some time or visit a hobby shop. Just the number of retail hobby shops catering to this market in any metropolitan area will give you a feel for the size of the market.

OK. The task then becomes finding something that mobile robots can do that is entertaining and exciting to watch. We need something where a hobbyist can spend a relaxing afternoon running his/her robot, alone or with others, and the machine should cost no more than \$500. All of the R/C model categories fulfill these requirements, but as of now only robot-builder GearHeads (like me) derive relaxation from fiddling with robots. True GearHeads don't mind homebrewing everything, are tolerant of their robots' shortcomings, and are optimistic beyond reason.

Most folks, however, need something easier to get into and more solid to enjoy. In order for a robot-as-hobby market to take off, they would need plug-n-play robot building block modules that they could mix and match, but most important they would need something fun to do with their robots, something that R/C models can't do.

That something may well be Robot Sumo Wrestling, an emerging hobby/sport from Japan. The idea is simple (push your opponent out of the ring), but like the game of Go it has great depth. The rules are: robot must fit within a 20-cm-square footprint, no height restriction, and must weigh no more than 3 kg. You lose when any part of your robot touches outside the Dohyo (the 154-cm diameter Sumo wrestling ring). No flame throwers, no coating your opponent with honey, no chip-zapping tesla coils, and so forth. It's purely a contest of speed, strength, and strategy.

A beginner can start with very basic hardware yet still achieve something worthwhile, while the possibilities for advanced hardware and software are immense. There have been several All Japan Robot Sumo Wrestling Contests. The latest one was in Tokyo December '92 where there were over 600 robots entered. Having been in the amateur robotics trenches for over fifteen years, let me tell you, that is a STAGGERING number of robots to be

gathered in one place. The event gets prime-time coverage on Japanese TV, the winners are treated like rock superstars. The atmosphere is reminiscent of the MIT engineering student competitions that have become so popular the last several years. Top prize in the '91 competition was 1 million yen, about \$7700 at the time.

The important thing, though, is these robots are great fun. I've viewed a couple of hour-long videotapes of the '91 and '92 contests (all in Japanese, which I don't speak), and let me tell you they are a kick in the butt to watch. I think a lot of other people will agree once it becomes established in this country, and once that happens, we will have our first large mobile robot market. That will get the whole market jump started.

Comments invited..

**Msg#:11404**

**From: JIM WHITE To: ROBERT NANSEL**

The R/C hobby market is a very good model to look at. The building that is done is strictly mechanical stuff. The number of electronics-oriented hobby building is very limited (I know, I'm in that tiny group). Go in to a hobby

## PLIX™ HARDWARE X-10™ TRANSCIVER CHIP

Micromint introduces its new Power Line Interface for X-10. Plix is an 18-pin ASIC chip that automatically handles all the specialized X-10 timing and bit-shuffling between a computer and a TW523 power line module.

- ✱ Complete interface between parallel port and TW523 or PL513 modules
- ✱ Performs all X-10 transmit and receive functions
- ✱ Detects AC power loss
- ✱ Simple interface and timing - operates even in interpreter BASIC
- ✱ Low power - only 1.8 mA @ 5 V

PLIX chip and  
data sheet

**\$20**

100 qty. OEM

**\$12**

**PLUS SHIPPING**

Call **1-800-635-3355**

or Write for a Plix data sheet

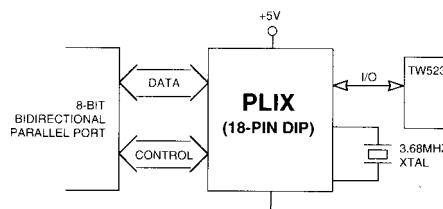
### PLIX EVALUATION BOARD

To speed your evaluation, we have produced a low cost Plix interface that plugs into a parallel printer port. The Plix-EKit includes a Plix chip and data sheet, PCB with all components, application note, and PC-compatible software on diskette.

**\$29.00**

**PLUS SHIPPING**

Add \$30 for TW523 X-10 transceiver module



**MICROMINT, INC.**

4 PARK ST., VERNON, CT 06066  
(203) 871-6170 FAX: (203) 872-2204

inc  
X-10 is a trademark

# CONNECT TIME

shop (at least in southern California) and most will be dominated by parts for the several very popular lines of R/C cars. Those that support flying and boating hobbies will have kits and parts for those models. Electronics are strictly off-the-shelf.

A good idea how big a barrier do-it-yourself robots are in the area of programmable R/C equipment. The high-end R/C transmitters are all microprocessor based these days, and allow various degrees of programmability, from basic channel mixing to units with PC interfaces. The vast majority of the R/C hobbyists consider this equipment "too complex" to use. The programming knowledge and skills required for this equipment is less than what is required for even basic robotics.

I am not saying there is no hope. Just that the learning curve will be very steep for the "general public," and the transition from R/C to autonomous systems will be very gradual for the bulk of the market. This is not a big problem because the basic "technology" and market channels to distribute the products is the same. The economies of scale for the combination of both types of robots will benefit both. The buyers will choose whichever they prefer.

---

## FCC testing-Designing for compliance

**Msg#:10750**

**From: ROBERT LUZENSKI To: ALL USERS**

I'd like to hear comments and advice from anyone who has been through the FCC process for computing devices. Also, if anyone knows of any good references to articles and books on the subject I'd be interested. I've called the FCC BBS on the subject, and that was a good start, but I think comments from real people who've been through it would be helpful to me and others.

**Msg#:10769**

**From: JIM WHITE To: ROBERT LUZENSKI**

I just went through my first time at having to have my designs get regulatory approval. FCC (Part 15 and Part 68), UL, DOC (Canadian FCC), and CSA (Canadian UL). I have two bits of advice.

First, design for compliance from the beginning. For FCC Part 15 this means creating a quiet design. My system has two parts, a hand-held terminal and a cradle (which includes a modem). This was a typical super rush, so the "important" part got lots of attention [the terminal] and the cradle didn't get the time it needed during the prototype phase. For cost reasons the cradle was dictated to be two-layer PCB, which meant no power or ground planes, and for time reasons the layout was just slapped together. As a

result, this innocent-looking little board with a modem chip set and an 80C51-type CPU on a 7.372MHz crystal was a real screamer! When I took the prototypes in for initial emissions screening, the terminal didn't even show up on the 'scope, but the cradle failed hopelessly; both emitted and conducted radiation, even though both designs used the same crystal and type of CPU.

For the production version, I naturally focused lots of attention on layout, creating large copper pours for power and for ground. When I took the new design in for screening, it disappeared from the 'scope also (except for teeny-tiny blips at the crystal fundamental and third harmonic). With both units so far below the emission limits, I will not have to get certification for Part 15 (I forget what they call it, but if your product doesn't emit [meaningful] radiation you don't have to get a certificate).

The moral of the story is, use good decoupling practices and get your prototypes (or whatever version first looks close to production in terms of layout and fabrication) screened as soon as possible. Don't wait until your design is "done," then try to get certified. Many people do, and it can be done, but the results are terribly costly (shielding, sprays, beads, chokes, etc.).

My second bit of advice is get a good lab to do the work for you. I do not recommend going to a service that deals primarily in the paperwork, and leaves the testing issues up to third parties. The process is already time consuming enough without that hassle as well. I picked the company I used because they are 30 minutes from the office and because they are a full-service "turn-key" compliance lab. For safety certifications (UL/CSA/VDE) they do the testing in their own labs, then get the results approved by the agency (which makes on-site visits, not just by the mail). Many compliance "labs" are really consultants who do the paperwork and deal with third parties for testing. This is especially common for safety certification because you can fill out the forms then send your stuff to the agency (e.g., UL) who then does the testing.

---

## Buried guide wire

**Msg#:10209**

**From: KEITH DONADIO To: ALL USERS**

I want to bury an insulated wire about four inches in the ground and have a toy electric car follow the wire. The maximum total distance between the sensor and the wire will be 10 inches.

I thought this would be easy and tried using a 1-kHz oscillator and a small coil of wire as the detector which fed an op-amp. I didn't have good results.

# CONNECTIME

Should I use a sine wave or a square wave? Audio or RF? Tuned or untuned?

**Msg#:10259**

**From: JAMES MEYER To: KEITH DONADIO**

If the buried wire isn't in the form of a large, open loop that lets you put current into it by connecting to both ends, then you should consider configuring it as such. You need to get some current flowing in the wire if you want the coil to pick it up easily. The frequency isn't \*too\* critical. The higher it is, the smaller you can make the car's sensor coil. A sine wave will be just fine. I'd guess that 100 to 500 kilohertz would be just about optimum.

The small sensing coil should be tuned to the same frequency that you feed to the buried loop.

Once you get the coil to detect the buried wire, how are you going to let the car decide whether it needs to turn left or right to get back over the wire?

**Msg#:11103**

**From: PELLERVO KASKINEN To: KEITH DONADIO**

Generally speaking, you would not want to bury two wires where one will do an adequate job. Keeping the distance between the two wires tends to become an exercise in futility if the soil is "normal." I think we are talking about near-field attenuation conditions in your application and that should make some simple differential principles at the receiving side practical. The reason why I say this is because in LVDTs (Linear Variable Differential Transformers) used for position measurements, there is a definite benefit in using the phase relationship instead of amplitude relationship. But there you have the generator signal available for the phase reference, which is much more difficult to implement in the buried guide wire case.

Anyway, the principle should be that of two pickup coils in a half-bridge configuration. Your coils could be large, air-core loops or wound over a ferrite antenna rod. I think the antenna rod construction is easier to make in a matching set.

A plain 60-Hz looks attractive for the signal, but may bring in some surprises near other 60-Hz lines. I remember a sailing boat speed measuring device that I helped the boat owner to repair and calibrate. I got on my first sailing boat trip as a result. And the speed indicator worked fine throughout the trip. Due to high winds toward the end of the two-day sailing, we were approaching or even slightly exceeding the design speed of the boat. Then, all of a sudden, the indication more than doubled, then came back to normal. Looking at the shore to both sides, I saw the warning signs of an undersea power cable. The magnetic field from the cable overpowered the feeble field of the permanent magnet in the sensor propeller..

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.

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.

**I R S**

428 Very Useful

429 Moderately Useful

430 Not Useful

## ATTENTION SUBSCRIBERS

Please direct all subscription requests and questions to our **NEW** Customer Service Department:

**THE** COMPUTER APPLICATIONS  
JOURNAL

P.O. Box 7694  
RIVERTON, NJ 08077

**(609) 786-0409**

# STEVE'S OWN INK

## The Club

Q

he obvious purpose of a magazine like the *Computer Applications Journal* is to communicate new ideas and designs to an audience of receptive readers. In this age of CD-ROMs containing hundreds of volumes of resource material, printed magazines can appear anachronistic. But don't let appearances be misleading. It is not the quantity of information that is important, it's the quality and timeliness of the details that are significant.

Probably one of the most gratifying aspects of our relationship with the readership is that the information exchange has become a two-way street. Technical magazines are usually a one-way information source where editors and authors provide snippets of wisdom and then vacate the scene until the next issue. Since the staff of CAJ consists of working engineers, we have the same responsibilities as you to produce real results and can't just talk about technology as something left as an exercise for the reader. We seek new information with the same zealously as you.

Believe it or not, the single greatest source of ideas and resources for the *Computer Applications Journal* has become the engineers and scientists who regularly read the magazine and hang out on our BBS. I use "hang out" as a term of endearment and not disdain. Our BBS has become a club where the only rule is a devout dedication to the exchange of technical knowledge.

What is most amazing is the breadth and depth of the discussions that go on among the participants. Fully half of the magazine subscribers have called the BBS, and with more than 70,000 calls a year there are lots of people involved. At any one time we might have conversation strings about GPS satellite receiver interfacing, the physical formula and material source of the goop in lava lamps, using laser interferometry, patient position sensing in a CAT scanner, linear displacement transducers, and making tactile sensors for robotics.

More than the topics themselves, the level of expertise of the callers to the BBS is astounding. Someone will ask something as insignificant as, "Can anyone explain how the fuel injection in a Ford differs from the Bosch design in a Porsche?" and, somewhere in the abundance of technical answers will be, "In my patent for that particular Ford device, it...." I'm amazed how many times the actual inventors and designers end up participating in these conversations.

What I appreciate most is that the participants of the BBS treat membership as an exclusive club. The dues are that they share what they know when a question overlaps with their expertise. Such interactive involvement is not legislated, merely encouraged. There isn't a day that even I don't learn something new from some contributor.

So when you look at your next issue of CAJ, consider that the brains behind it don't come from just an editorial staff. Reflect instead on a collective product from an insanely dedicated group of individuals committed to the proposition that knowledge is something to be communicated and shared rather than classified and concealed. There are some really brilliant minds out there. I feel privileged that so many of them are willing to call us and share their wisdom.

