HA&BC

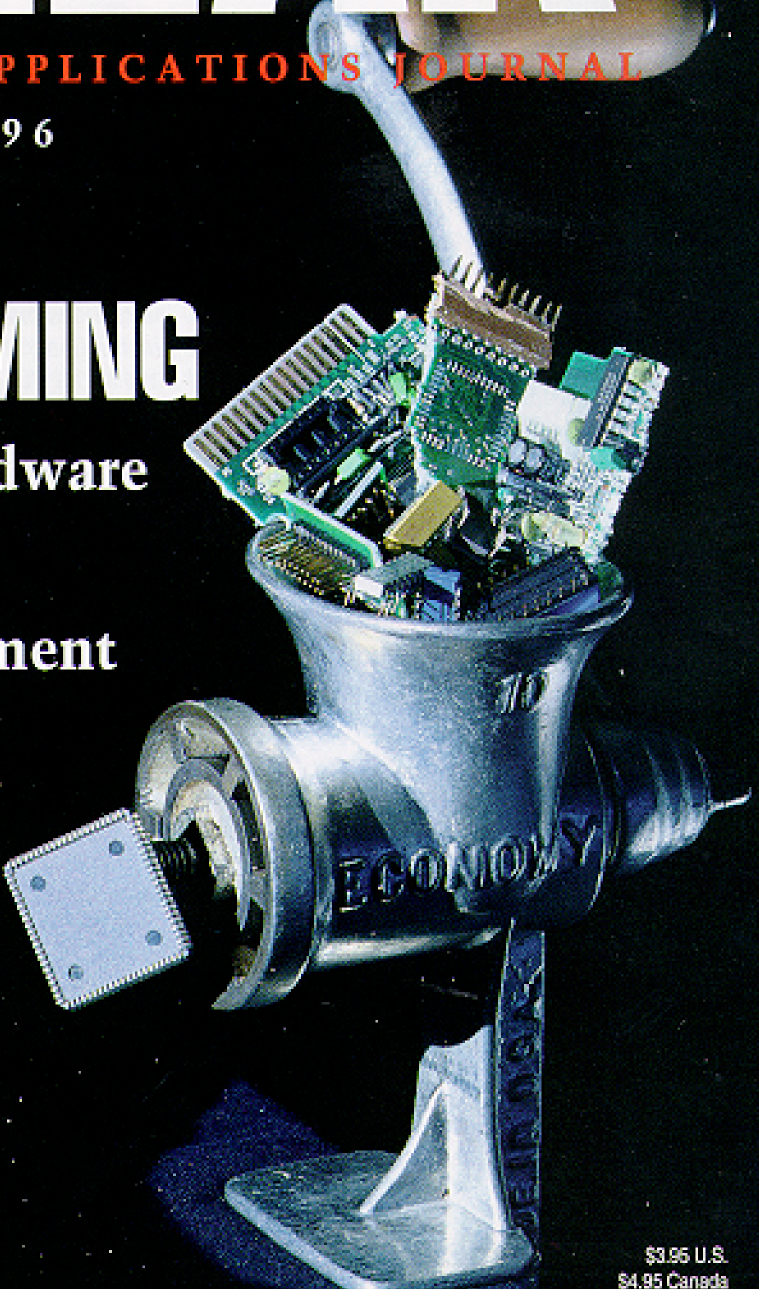
CIRCUIT CELLAR INK®

THE COMPUTER APPLICATIONS JOURNAL

#74 SEPTEMBER 1996

SPECIAL BONUS SECTION: HOME AUTOMATION & BUILDING CONTROL

EMBEDDED PROGRAMMING



Getting to the Hardware in Windows 95

Biomedical Instrument Interfacing

USB—Solving the Rat's Nest

HABC: Intelligent X-10 Control

09
0 74470 75349 0

$3.95 U.S.
$4.95 Canada

TASK MANAGER

The Windows 95 Convert

So, have you tried Windows 95 yet? I know more than a few technical people who are scared to death to even open the box. Their DOS or Win 3.1 machines are chugging along just fine, doing what they need them to do, day after day. Why flirt with disaster?

I have two machines on my desk. I use my Mac for graphical- and magazine-oriented tasks and my PC for text-based tasks and code development. Until recently, my PC lived primarily on DOS 5.0 with occasional forays into Win 3.1, but given the minimal spare hard-disk space (on average, 3–4 MB), I couldn't consider any more.

As more and more applications have been coming out for Windows, I saw the need to upgrade. Adding a (much!) larger hard drive and a CD-ROM drive, I bit the bullet, put the Win 95 CD-ROM into the drive, and unleashed it. What I ended up with was a pleasant surprise—a stable, easy-to-use, and peppy machine that has increased my productivity markedly.

I've since installed Win 95 on four other machines and am pushing to make it our company-wide standard. It's a real bummer to have to use a Win 3.1 machine now.

Several years ago, Ed wrote about doing code development under OS/2. I've been able to create a similar setup under Win 95 that greatly enhances my development environment. In one window, I have my text editor always open. I press a button and launch the DOS-based assembler in another window. I then load the assembled code into the ICE in yet another window. Finally, I have a comm program open in another window that I use to communicate with the microcontroller running the code.

I can quickly switch back and forth among all the windows, giving commands to my application through the comm program and watching the effect in the ICE window. Under DOS, I had to do each step sequentially, quitting each program and restarting the next each time. I used a separate computer altogether to run the comm program (since I couldn't run both the comm program and the ICE front end simultaneously).

Many of you may dismiss Windows 95 as a crutch necessary only for novices that can't handle the rigors of DOS and think it can't be used in a serious embedded development environment. Rethink your position. I don't think you'll be disappointed.

I mention Win 95 because one of our feature articles this month shows some tricks that can be used by Win 95 programmers for getting directly to hardware projects. It really is possible to get around Win 95's blanket of protection when necessary.

Among our other features are an introduction to the merits of EPLDs, selecting and designing with high-resolution A/D converters, Part 1 of a series discussing design issues in biomedical instruments, and the final installment covering in-circuit emulators.

In HABC, we look at adding some intelligence behind X-10 to make it quicker and more reliable, and we get a behind-the-scenes tour of one homeowner's wiring infrastructure.

Finally in our columns, Ed continues with his zero-beat tuner, Jeff monitors power usage in his home, and Tom explores the up-and-coming USB standard.

editor@circellar.com

CIRCUIT CELLAR INK®

THE COMPUTER APPLICATIONS JOURNAL

FOUNDER/EDITORIAL DIRECTOR
Steve Ciarcia

EDITOR-IN-CHIEF
Ken Davidson

MANAGING EDITOR
Janice Marinelli

TECHNICAL EDITOR
Elizabeth Laurençot

ENGINEERING STAFF
Jeff Bachiochi & Ed Nisley

WEST COAST EDITOR
Tom Cantrell

CONTRIBUTING EDITORS
Rick Lehrbaum
Fred Eady

NEW PRODUCTS EDITOR
Harv Weiner

ART DIRECTOR
Lisa Ferry

PRODUCTION STAFF
John Gorsky
James Soussounis

CONTRIBUTORS:
Jon Elson
Tim McDonough
Frank Kuechmann
Pellervo Kaskinen

PUBLISHER
Daniel Rodrigues

PUBLISHER'S ASSISTANT
Sue Hodge

CIRCULATION MANAGER
Rose Mansella

CIRCULATION ASSISTANT
Barbara Maleski

CIRCULATION CONSULTANT
John S. Treworgy

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 (860) 875-2751. Periodical rates paid at Vernon, CT and additional offices. One-year (12 issues) subscription rate U.S.A. and possessions $21.95, Canada/Mexico $31.95, all other countries $49.95. All subscription orders payable in U.S. funds only, via international postal money order or check drawn on U.S. bank.
Direct subscription orders and subscription related questions to Circuit Cellar INK Subscriptions, P.O. Box 698, Holmes, PA 19043-9613 or call (800) 269-6301.
POSTMASTER: Please send address changes to Circuit Cellar INK, Circulation Dept., P.O. Box 698, Holmes, PA 19043-9613.

Cover photography by Barbara Swenson
PRINTED IN THE UNITED STATES

For information on authorized reprints of articles, contact Jeannette Walters (860) 875-2199.

HAJAR ASSOCIATES **NATIONAL ADVERTISING REPRESENTATIVES**

NORTHEAST & MID-ATLANTIC	**MIDWEST & SOUTHEAST**	**WEST COAST**
Barbara Best	**Christa Collins**	**Barbara Jones & Shelley Rainey**
(908) 741-7744	(954) 966-3939	(714) 540-3554
Fax: (908) 741-6823	Fax: (954) 985-8457	Fax: (714) 540-7103

Circuit Cellar BBS—24 Hrs. 300/1200/2400/9600/14.4k bps, 8 bits, no parity, 1 stop bit, (860) 871-1988; 2400/9600 bps Courier HST, (860) 871-0549. World Wide Web: http://www.circellar.com/

INSIDE ISSUE 74

READER I/O

NO MORE MIDNIGHT STUMBLES

I have a simple solution to a common household problem, and I'm looking for someone with more hardware background to steer me in the right direction.

In the U.S., we often have a bedside lamp plugged into a wall outlet and controlled by a switch at the door. So, you can read in bed, turn off the bedside lamp, and go to sleep.

The problem comes the next night. You come to the door, it's dark, and the light switch at the door is already on. To make the bedside lamp come on, you have to cross the room in the dark to turn the lamp switch.

My solution has an electronic switch in the lamp which detects that mains power has gone from on to off and reverses the state of the electronic switch. Mains power going from off to on doesn't affect this switch.

If the lamp has my Smart Lamp Switch, you turn it on from the door switch by on-off-on. I have a Patent Pending on this idea, and my friends love the prototypes. I want to find a company to manufacture it in quantity. If you're interested, let me know. I'll send you a full description and a copy of the Patent Office receipt.

Neil Bennett
neilb@halcyon.com

Contacting Circuit Cellar

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

Mail: Letters to the Editor may be sent to: Editor, Circuit Cellar INK, 4 Park St., Vernon, CT 06066.

Phone: Direct all subscription inquiries to (800) 269-6301. Contact our editorial offices at (860) 8752199.

Fax: All faxes may be sent to (860) 872-2204.

BBS: All of our editors and regular authors frequent the Circuit Cellar BBS and are available to answer questions. Call (860) 871-1988 with your modem (300–14.4k bps, 8N1).

Internet: Letters to the editor may be sent to **editor@circellar.com.** Send new subscription orders, renewals, and address changes to **subscribe@circellar.com**. Be sure to include your complete mailing address and return E-mail address in all correspondence. Author E-mail addresses (when available) may be found at the end of each article. For more information, send E-mail to **info@circellar.com**.

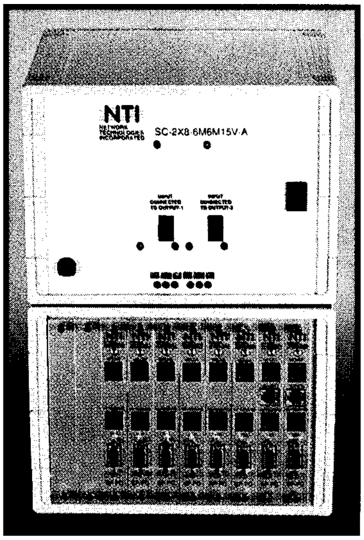
WWW: Point your browser to **http://www.circellar.com/**.

FTP: Files are available at **ftp://ftp.circellar.com/pub/circellar/**.

NEW PRODUCT NEWS

Edited by Harv Weiner

MULTIPLE PC ACCESS



Network Technologies introduces **SC-2X8-6M6M15V-A**, a two-user, eight-PC keyboard, monitor, and mouse switch. The switch maintains and updates networks, provides technical backup to computer presentations, and offers an extra point of access to multiple computers.

Both sets of monitors, keyboards, and mice connect directly to the unit's input ports and can be placed 250′ away from the switch. Both users can work on different PCs simultaneously, allowing more flexibility than single-user switches. The automatic-reboot circuitry simulates the presence of a keyboard and mouse to all attached PCs. This allows all eight PCs to boot error free, although only two keyboards and PS/2 mice are present. PCs are selected and controlled using keyboard commands or the optional hard-wired remote control.

The video connector is a 15-pin high-density D, while both keyboard and mouse connectors are 6-pin mini-DIN. The 150-MHz bandwidth supports 1600 x 1200 video resolution with no degradation. This switch is compatible with all models and variations of PS/2 computers, as well as VGA and SVGA video, PS/2 keyboard, and PS/2 mouse.

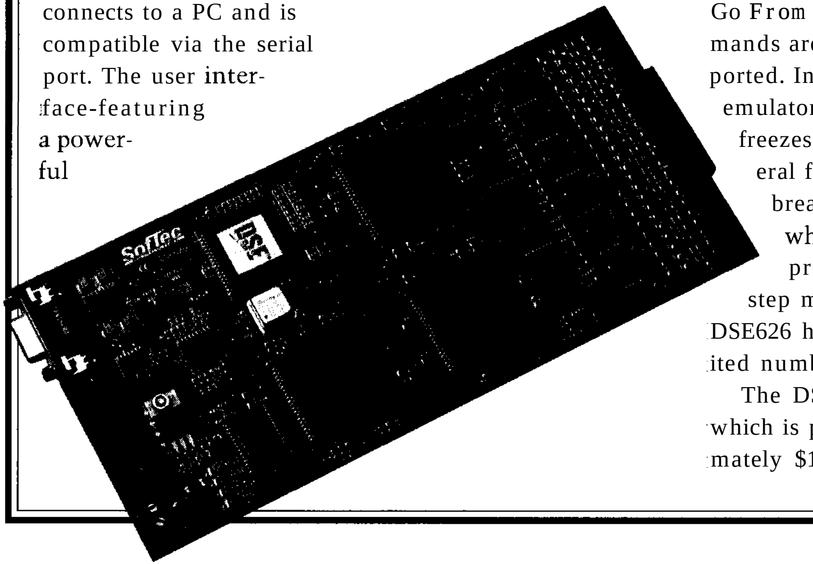
The SC-2X8-6M6M15V-A is housed in an 8.8" x 7.8" x 6.5" plastic case with a front-panel lighted display. It is powered by 110 or 220 VAC at 50 or 60 Hz. The unit comes with a one-year warranty and retails for $3630. Optional remote controls are $350 per console.

Network Technologies, Inc.
1275 Danner Dr. • Aurora, OH 44202
(216) 562-7070 • Fax: (216) 562-1999

#500

IN-CIRCUIT EMULATOR

SofTec Microsystems announces the DSE626 In-Circuit Emulator, a low-cost, PC-based real-time development system for SGS-Thomson ST6260 and '65 microcontrollers. DSE626 offers real-time, transparent emulation at the maximum frequency of the ST6 family. DSE626 connects to a PC and is compatible via the serial port. The user interface-featuring a powerful debugger-runs under Microsoft Windows.



DSE626 software features a state-of-the-art windowed interface with pull-down menus, dialog boxes, function and hot keys, mouse support, and hypertext help. All information regarding the microcontroller and the user application is available on-screen. It provides separate windows for examining source code, code memory, data memory, watch variables, special registers, program stack, and breakpoints. Every microcontroller resource can be directly edited from the appropriate window.

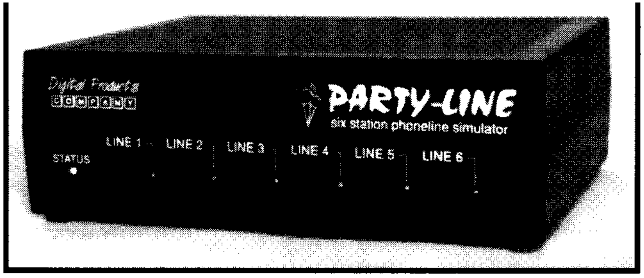
Reset, Step (Into, Over, Out, Multiple), and Run (Go, Go From Reset, Go From New PC) commands are currently supported. In addition, the emulator stops (i.e., freezes) core and peripheral functions after a break acknowledge or when executing programs in step-by-step mode in idle time. DSE626 handles an unlimited number of breakpoints.

The DSE626 package, which is priced at approximately $1000, comes with the emulator unit, Windows software, serial cable, 20- and 28-pin probes, user guide, and a one-year warranty.

SofTec Microsystems
viale Rimembranze, 19C
33082 Azzano Decimo
Italy
(t39) 434-640113
Fax: (t39) 434-631598
softec@mbox.vol.it
http://www.vol.it/softec/

#501

NEW PRODUCT NEWS



PHONE-LINE SIMULATOR

Party-Line is a six-station telephone-line simulator with Caller ID. It accurately generates North American call-progress sounds such as dial tone, busy signals, and ringback. It also transmits Caller-ID data and supports distinctive ringing services.

Party-Line can test or demonstrate standard telephones, answering machines, fax units, voice-mail systems, or modems. The talk circuit provides an exceptionally clear voice path and supports data communications to 28.8 kbps.

Telephone equipment used with Party-Line behaves as if it's connected to a real phone line. The user hears the same familiar dial tone, ringback, and busy signals. DTMF (touch tone) dialing is used to make calls and control the system features.

You call any of six extensions by dialing seven-digit local numbers or ten-digit long-distance numbers. A three-digit speed-dial mode supports convenient intercom dialing. Area-code and number prefixes as well as extensions can be assigned.

The Caller-ID feature can transmit single- or multiple-message signaling methods. An internal real-time clock simulates the time and date. All popular message-delivery codes, including "number blocked" and "out of area," are provided. Dialing codes enable Caller-ID features, including deliberate broadcasts of Caller-ID data-transmission errors.

An internal 20-Hz ring generator ensures nearly any telephone gear can be called. The ring circuit uses a superimposed voltage source for compatibility with popular telephone products.

Party-Line is offered as a kit and comes complete with a high-quality PCB, all electronic components, and a X-page technical manual. The kit sells for $199.95 and an optional custom enclosure is $32.95. Fully assembled and tested units are $425.

Digital Products Co.
134 Windstar Cir.
Folsom, CA 95630
(916) 985-7219
Fax: (916) 985-8460
DigProd@aol.com

#502

APPLICATIONS CODE GENERATOR

The MP-Driveway application code-generator tool supports Microchip's PIC-16Cxx and PIC 17Cxx S-bit microcontroller families. Users can produce tested and documented C, most of which controls on-chip peripherals.

This Windows-based software eliminates the need to code peripheral functions and learn the specific features of supported chips. You gain full hardware-software integration. Using MP-Driveway, developers can generate device drivers, application modules, initialization routines, complete test functions, the main template, interrupt service routines, fully documented code, and examples of function usage.

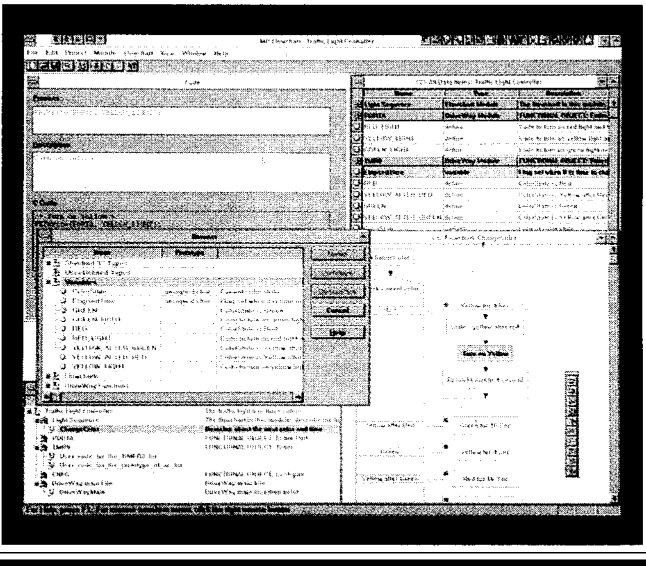
MP-Driveway enters device-driver parameters by calculating complicated values based on entered requirements or providing a dialog window that enables easy entry of all required values. An online interactive datasheet provides hypertext data on PIC 16 and '17 microcontrollers and peripherals, modes and registers, and pins.

MP-Driveway software consists of a user-interface shell and a knowledge base. The interface includes screens and operation menus for defining chip operation, setting initial values, and generating source code. The knowledge base includes the components required by the interface to generate a device driver for the supported chip.

MP-Driveway requires an 80386-based PC or greater, MS Windows 3.1 or higher, 4-MB RAM, 5-MB free disk space, and a 3.5" disk drive. It lists for $495.

Microchip Technology, Inc.
2355 W. Chandler Blvd.
Chandler, AZ 85224-6199
(602) 786-7200
Fax: (602) 899-9210
http://www.microchip.com/

#503

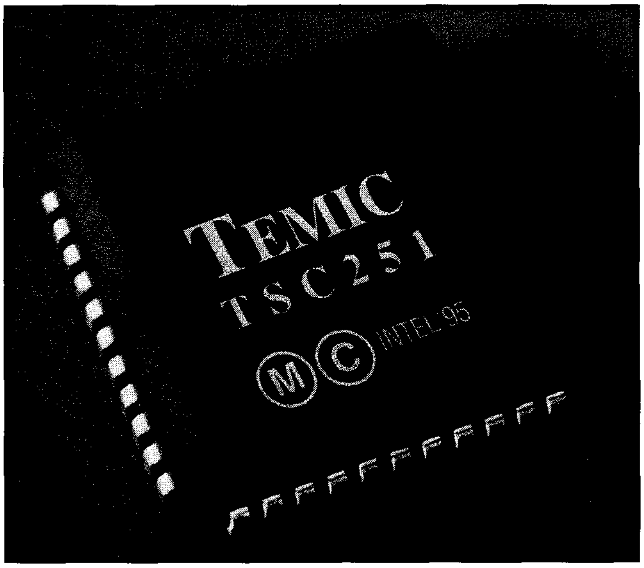


NEW PRODUCT NEWS

80251 DERIVATIVE MICROCONTROLLER

The **TSC80251A1** is the first derivative of the advanced 80251 8-bit microcontroller family for multiple applications in automotive, telecom, computer, broadcast-media, and industrial applications. Available in both 3- and 5-V versions, the chip is equally suited for other demanding systems such as communications equipment and computer peripherals.

The chip features 24 KB of OTP, EPROM, or mask ROM (ROMless versions are also available). It provides 1024 bytes of RAM with 256 KB of external memory space and up to 16 MB of memory space for code and data. The chip also integrates a programmable counter array, as well as three pulse-width measurement units.



With C51 binary-code compatibility, the 8025 1 architecture enables designers to improve performance by a factor of five without any change to their existing code. By using the 8025 1 instruction set, performance can be improved by a factor of 15. Significantly increased throughput (compared with 80C5 1 products] means designs with the TSC8025 1 can use drastically reduced clock frequencies. This reduction means lower power consumption and EMC with no corresponding loss in performance.

The TSC80251 family significantly reduces required code with dual 16-/32-bit instructions, including 16 x 16 multiplication and division. Optimized with its larger stack and new instructions to support C language, designs using the chip require less development time. The ROMed version of the chip sells for $9 in lO,OOO-piece quantities. The EPROM version sells for $29 in 1000-piece quanitities.

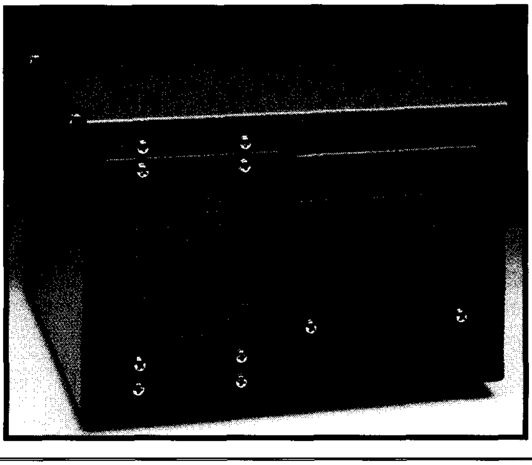
Temic
2201 Laurelwood Rd. • Santa Clara, CA 95054 • (408) 567-8220 • Fax: (408) 567-8995

#504

PC CHASSIS

The Model 9040 Industrial Embedded Chassis is ideal for users requiring a ruggedized chassis in harsh, space-conscious, unattended applications. Typical applications include factory automation, telecommunications, data acquisition, process control, medical instrumentation, and intelligent terminals.

The Model 9040 includes an ISA-compatible multilayer passive backplane designed for use with standard plug-in CPU, display, and I/O adapter boards. The backplane supports up to four three-quarter-length PC/AT plug-in boards. The drive bay supports up to two half-height or one full-height 3.5" drive devices. Each drive can be individually shock mounted. A 32.4-CFM cooling fan with removable filter is provided for the main chassis and circuit-board areas. The unit includes a 70-W autosense switching power supply.



The Model 9040 is housed in rugged 16-gauge reinforced steel measuring 9" x 5.8" x 8.8". It comes complete with a user manual and a two-year warranty. The Model 9040 sells for $355.

Micro Computer Specialists, Inc.
2598 Fortune Way
Vista, CA 92083
(619) 598-2177
Fax: (619) 598-2450
mcsi@mcsi1 .com
http://www.industry.net/mcsi/

#505

NEW PRODUCT NEWS

EMBEDDED BIOS

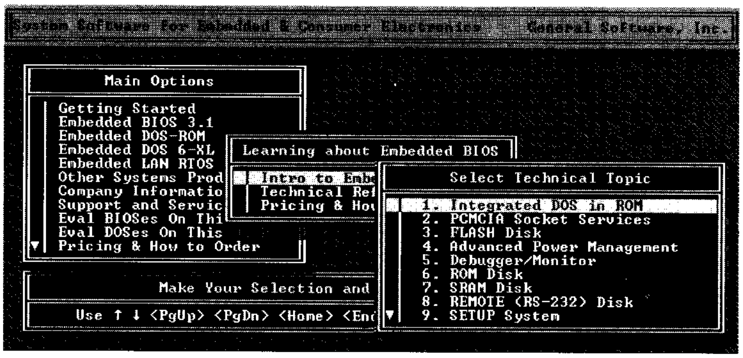
General Software's **Embedded BIOS 486-SX/GX** supports Intel's new '486-SX/GX low-power microprocessor. It takes advantage of the new microprocessor's features to reduce power consumption and increase performance in consumer electronics commodity products (e.g., cell phones, pagers, and personal Internet communications devices).

The BIOS provides system support at all stages of a product's life cycle. For example, during prototyping, the built-in ROM debugger makes it easier to ensure that memory-refresh circuitry, VLSI chipsets, and PCMCIA controllers are connected properly.

During development, the Embedded BIOS 486-SX/GX remote disk allows the target to share the disk drives of a development PC or host. It's unnecessary to burn EPROMs for testing the application on the target.

During manufacturing, in quality assurance, and in the field, the special manufacturing mode enables remote hardware or even a custom test jig to program flash, test components and software, and update system and application software.

OEM adaptation kits with full source code are available for **$4995**. Complete with its own run-from ROM DOS called Mini-DOS, the total BIOS/DOS royalty starts at **$4** per copy.



General Software, Inc.
320 108th Ave. N.E.,
Ste. 400
Bellevue, WA 98004
(206) 454-5755
Fax: (206) 454-5744
general@gensw.com
http://www.gensw.com/general/

#506

SPECIAL PRICING!
12 VDC MINI-FAN

Sunon # KD1206PTS1 12Vdc, 2.2 W fan. 60mm X 60mm X 25mm. Black plastic housing. Five blade impeller. 18" pigtail leads. While Supply Lasts!

$4.00 each **CAT# CF-35** **10 for $35.00**

12 POSITION BARRIER STRIP

20 amp, single row, pc mount barrier strip with screw-clamp terminals. Just slip wire under clamp and tighten it down. 12 terminals on 0.375" centers. Black thermoplastic strip is 4.56" X 0.61". Barriers between screws are 0.63" high. Large quantities available.

65¢ each **100 for $50.00 • 1000 for $300.00** **CAT #TER-124**

SET OF 12, SPDT NUMBERED
LIGHTED SWITCHES

Twelve push-on/push-off switches (only 6 pictured). Each switch has a 0.6" square yellow lens with a 0.25" high black numeral 1 through 12. They are lighted by a 28 volt, #85 wedge base incandescent lamp which can be easily replaced if a different voltage is desired.

CAT #PBL-12 **$15.00 per set of 12**

10K TEN TURN
PRECISION POT

BI Technologies #8146 10K ohm linear taper, ten-turn precision pot. 10% resistance tolerance. 0.25" diameter X 0.5" long shaft. 0.37" (3/8") diameter threaded bushing. 0.88" diameter X 0.75" body.

$4.50 each **CAT# PPT-10K** **10 for $40.00**

THERMOELECTRIC COOLER
PELTIER JUNCTIONS

NEW LOWER PRICING

Current applied to the device will produce heat on one side and cold on the other side, up to 68° C difference between the two sides. Modules can be mounted in parallel to increase the heat transfer effect or can be stacked to achieve high differential temperatures. 127 thermocouples per device. Operates on 3-12 Vdc. Requires a heatsink to prevent overheating. *Two sizes available.*

1.18" (30 mm) square X 0.15" (3.8 mm) thick.
$17.00 each **CAT# PJT-1** 5 for $75.00

1.57" (40 mm) square X 0.15" (3.8 mm) thick.
$26.00 each **CAT# PJT-2** 5 for $110.00

Quantity Pricing Available!

ALL ELECTRONICS CORPORATION
INTERNET http://www.allcorp.com/

ALL ELECTRONICS CORP. • P.O. Box 567 • Van Nuys, CA 91408-0567
FAX (818)781-2653 e-mail - allcorp@allcorp.com

NO MINIMUM ORDER. Shipping and handling for the 48 continental U.S.A. $5.00 per order. All others including AK, HI, PR or Canada must pay full shipping. All orders delivered in CALIFORNIA must include local state sales tax. Quantities Limited. NO COD. Prices subject to change without notice.

ORDER TOLL FREE **1-800-826-5432**
Visa, Mastercard, American Express or Discover accepted

CALL, WRITE, FAX or E-MAIL for a FREE 64 Page CATALOG
Outside the U.S.A. send $2.00 postage.

#105

FEATURES

Getting Started with Xilinx EPLDs

FEATURE ARTICLE

J. Conrad Hubert

> The move from PLDs to EPLDs isn't a question of if, but when. In this article, Conrad shows you how to painlessly design with EPLDs. But, beware. If you're convinced, you might leave microcontrollers behind forever.

Programmable Logic Devices (PLDs) have steadily gained market share over discrete logic since the late 1970s. Simple PLDs—like PALs and GALs—are increasingly replaced by ever-larger complex PLDs, such as Erasable PLDs.

I'll show you that designing with programmable logic is as easy as with microcontrollers. But first, let's see if I can justify using PLDs based on the same criteria used to justify designing-in a microcontroller.

Are PLDs inexpensive? The equivalent of 800 gates can be purchased for $2.80 in 1000-piece quantities. Windowed EPLDs may be UV-erased and reprogrammed exactly like an EPROM. In production, you can use functionally identical plastic-encapsulated devices.

Regardless of cost, ease of use can save many labor costs. In comparison with wire-wrapping discrete ICs, designing with PLDs is positively a joy!

To show this, I had two challenges. I needed a simple example to illustrate how straightforward EPLDs are, and I wanted a real-world application to boast their enormous potential.

The first challenge was easy. I chose an electronic dice game. The second challenge was more problematic.

EPLD-based designs are usually proprietary. Their designers won't divulge the secrets. In other cases, examining the programmable logic outside the system isn't illuminating.

Then I spoke with David Rector at UCLA's Brain Research Institute. His application definitely wasn't trivial, and he could discuss it. He'll cover the use of a Xilinx EPLD in *INK 75* and 76.

Having handed off the second challenge, I'll give you some background on Xilinx's Field-Programmable Gate Arrays (FPGAs) and EPLDs—their similarities, differences, and when to use which.

EPLDs VERSUS FPGAs

Although Xilinx is best known as the originator of the FPGA, their 1992 acquisition of Plus Logic put them in the CPLD market. Plus Logic's old H2020 became the Xilinx XC7272. Xilinx's similar, but smaller, XC7236 completes the XC7200 family.

Xilinx's EPLD division designed an entirely new family-the XC7300 with I/O ranging from 38 in the XC73 18 to 156 in the XC73144. Packaging ranges from a 44-pin PLCC to a 225-pin BGA.

I'd never advise against designing in a Xilinx FPGA. But, as far as getting started is concerned, EPLDs are definitely the way to go.

The DS-550 software is an order of magnitude less expensive than their FPGA counterpart.

The DS-550 permits you to do what is called behavioral design (i.e., designs based on Boolean logic equations). Although you can design EPLDs with schematics, it costs about the same as Xilinx's FPGA tools. And, with their recent acquisition of NeoCAD, their PC development tools are Windows based. This makes a strong case for acquiring the DOS-based DS-550.

If you've been designing with PALs and GALs, Xilinx EPLDs are a natural progression. They are ideal for integrating the functionality of several smaller PLDs (e.g., 22V10s) into a single device. In fact, Xilinx specifies its EPLDs in terms of equivalent 22V-10s.

EPLDs are self-contained and instantly on (i.e., all configuration information is held in EPROM cells). Each time power is applied, the EPLD undergoes a self-timed configuration period lasting a few hundred microseconds.

During the configuration, data is read from the internal EPROM and into routing switches, giving the silicon its intended functionality. FPGAs [other than the XC8100 family) contain SRAM-based configuration memories (i.e., configuration data is held elsewhere in nonvolatile memory).

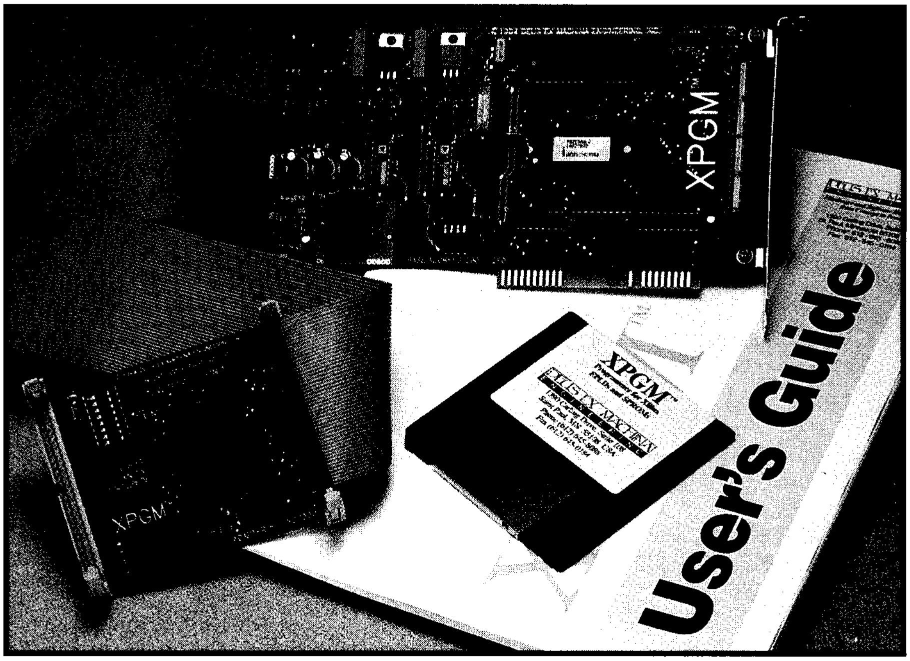


Photo 1—*XPGM* is a *Xilinx-certified programmer manufactured by Deus Ex Machina Engineering and distributed by Digi-Key.*

Since an FPGA's configuration memory is contained off-chip, it's simple to create a functional clone. Conversely, EPLD configuration data is on-chip. Cloning is difficult.

Like a microcontroller, an EPLD may be secured against unauthorized copying by programming a security bit or bits. But, it's still possible to selectively erase programmed security bits.

Xilinx has gone to great lengths to prevent such piracy in their XC7300 EPLD family. For selective erasure to be effective, the pirate must know the security-bit address.

Although various families of Xilinx FPGAs are specifically optimized, EPLDs offer a number of simultaneous advantages such as low propagation delay, high output-drive capability, and suitability for register-intensive tasks.

Because of smaller die sizes, EPLDs are generally faster than FPGAs. Since the XC7300 family provides a 24-mA output drive, they're natural for bus-interface tasks. Xilinx EPLDs are therefore common in PCI-compliant interfaces.

They are also useful for level translation since the core can operate at 5 V while its I/O buffers operate at 3.3 V. And, because EPLDs are register intensive, they are well-suited to applications like counters and state machines.

EPLDs force you to create synchronous designs in a way that FPGAs do not. It takes more self-discipline to create robust designs with FPGAs.

So, when does an FPGA make more sense than an EPLD? One selling point for FPGAs is that they can be rapidly reconfigured in-circuit. This is a boon to prototyping and can enhance the end product as well.

FPGAs are currently have more equivalent-gate capability than EPLDs. Very large designs typically require FPGAs. However, it's unlikely you can use 100% of an FPGA. As much as half of it may be unusable in any design. And, as an FPGA approaches full utilization, it becomes less likely that its pinout can be determined a priori.

Now you have the background; let me introduce the electronic dice game.

ELECTRONIC DICE

A six-sided die numbers from 1 to 6. The sum of two dice varies from 2 to 12, but not all sums are equally likely.

Although it seems complicated to generate the sum's properly weighted occurrences electronically, you only need a counter with 36 states.

The dice count speed is controlled by an RC oscillator. Since the logic simply steps through all 36 states, only

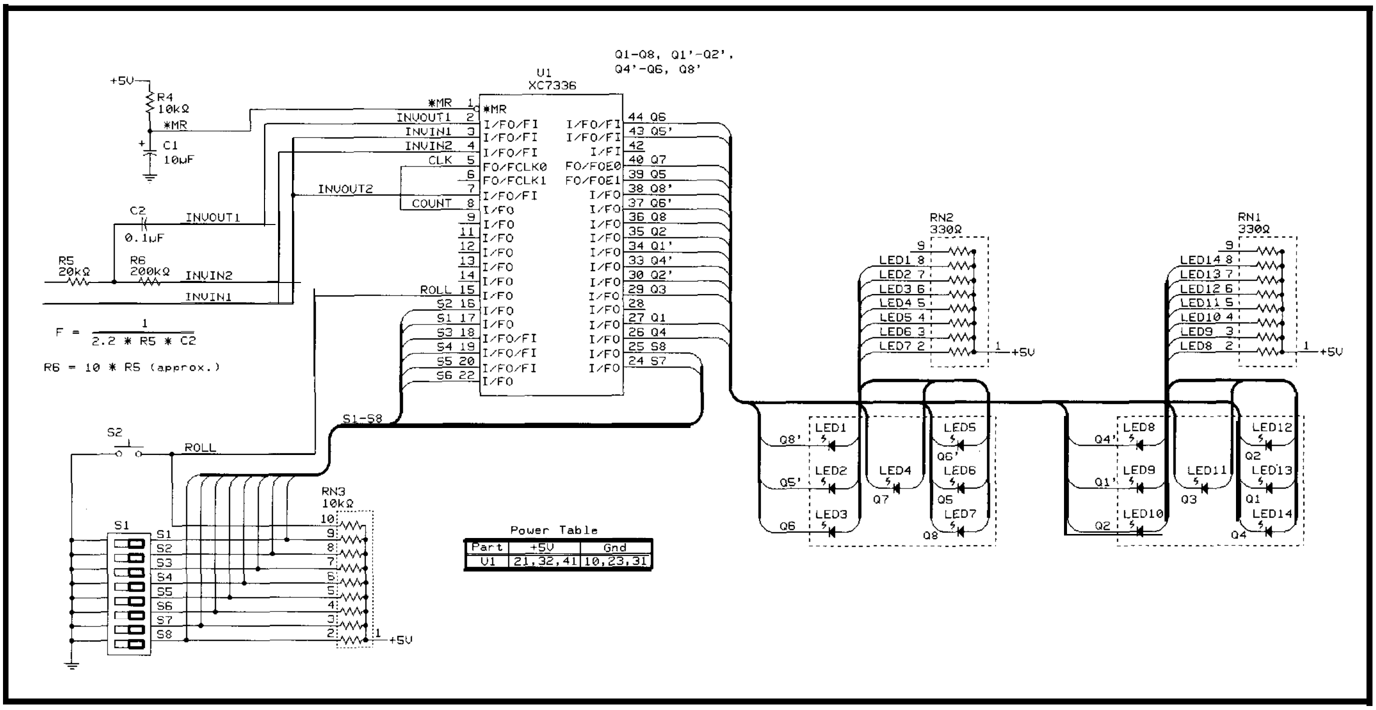


Figure 1--The circuitry for an electronic dice game is based on a Xilinx XC7336 EPLD

how long the roll button is pressed is random.

Typically, seven LEDs are driven by four bits since some are paired. However, each here is independent. Though inefficient, it offers greater flexibility.

Figure 1 shows the hardware for the electronic dice. The anode of each LED is tied to +5 V through a current-limiting resistor. An LED lights by bringing its cathode to ground [logic low).

At powerup, the EPLD has a self-timed configuration of -350 µs. R4 and Cl generate a power-up reset pulse. If the power-supply rise time is sufficiently fast (i.e., less than 5 ms), R4 and Cl can be eliminated, but you must tie the master reset pin to V,,.

Four EPLD I/O pins make two inverters. These, along with C2, R5, and R6, make up a crude RC oscillator.

This oscillator only suits the most undemanding timing applications. Its frequency is determined by the RC time constant of R5 and C2. Alter these values using the formula:

$$f = \frac{1}{2.2RC}$$

R6 isolates the RC network from the EPLD's input-protection diodes, and is not essential. If used, its value should be -10 times R5. When C2 = 0 1 µF and R5 = 20 kΩ, frequency is about 0.23 kHz.

The XC7336 requires its clock signals to be injected on pins 5 (FCLK0) and/or 6 (FCLK1). This explains why COUNT (output from gated osc.) is connected to the CLK pin (dice clock).

During initial testing, you may want to inject a 1-Hz square wave on

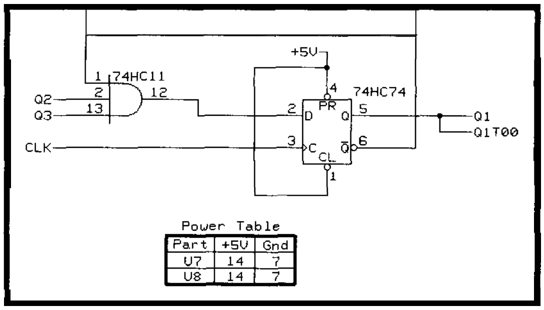


Figure 2—*The equivalent discrete-gate logic can further illustrate some of the* **behavioral equations** *found in* **Listing** *1.*

pin 6 to observe the state transitions. Counting should progress like: 1-1, 1-2, 1-3, …, 2-1, …, 6-5, 6-6, 1-1 . .

NAVIGATING THE DS-550

Xilinx's DS-550, a kind of assembler for silicon, processes design equations similar to those for PALs and GALs. After installing it, proceed with the dice game.

First, create the files DICE.PLD and DICE.VMF (see Listing 1 and Figure 3, respectively, or download the files from the Circuit Cellar BBS).

All .PLD files contain device equations. Let's examine some of the example logic equations in Listing 1.

q1.clkf = clk says the clock source for the q1 output is the clk pin. q1 := /q1 * q2 * q3 assigns q1 the logic state represented by (NOT q1) AND q2 AND q3 after the clock transition. q1too = q1 provides a redundant output to drive a second LED. Figure 2 shows the equivalent discrete logic circuit.

Figure 3 is a user-created file that assigns a name and function to a physical pin and implements pin freezing. Xilinx's pin-freezing is so good that they guarantee 100% utilization with pin freezing invoked.

Unless you have layout constraints, initially process your design without pin freezing, so the DS-550 can achieve optimal use of resources. Once a design is committed to a PCB, however, pin-freezing should be used for all subsequent design revisions.

To start the DS-550, execute XDM (XDM is a shell which stands for Xilinx Design Manager). After saving the part and family information, the software is ready for the equation file.

Enable -f (i.e., pin freezing) and -u (it drives unused I/O pads within the EPLD, so you can leave them externally unconnected). Various programs process input files. You get a lot of DICE files with various extensions.

At this point, you could use a third-party (e.g., OrCAD, ViewLogic, etc.) simulation tool to verify that your equations behave as expected. However, since DS-550 lacks an integrated simulation tool, you may have to resort to using the time-honored burn-and-crash method.

Alternatively, if you have a tool to simulate PAL designs, it can simulate and verify pieces of your EPLD design prior to running the DS-550.

Before proceeding to burn, look at the files created by the DS-550. The DICE.LOG file contains clues about what goes wrong if the DS-550 is unable to process your files correctly.

DICE.PIN lists device pin assignments. You always want the .PIN file to list the device's pins exactly as you defined them in the .VMF file (this is a good way to check that you remembered the pin-freezing option).

DICE.EQN shows how your equations are optimized and implemented. DICE.RES shows how the chip's resources were allocated.

Now, you're ready to make a programming file for the device. When you select DICE.VMH under the MAKE-PRG command, enter up to eight ASCII characters to help you identify what program is contained in a device.

The DS-550 now crunches the .VMH file. This process runs faster than the previous processes. The outcome has a .PRG extension and is readable by a device programmer. Now, you're ready to program a device.

GAINING FAMILIARITY WITH XPGM

XPGM, shown in Photo 1, is a command-line programmer for Xilinx devices. It uses one ISA slot inside your PC. A device adapter socket connects to a 50-pin ribbon cable which extends outside the PC. Each device package requires a unique adapter socket. The XC7336 requires Adapter 5.

Assuming you installed XPGM, the following steps will have you up and running quickly. Connect Adapter 5 to XPGM's ribbon cable, and insert a blank XC7336 into the ZIF socket.

The notched corner of the EPLD must align with the notch on the ZIF socket to properly orient the EPLD so that its pin 1 is opposite the silk-screened arrow on the adapter.

Now let's blank-check the device by entering xpgm -d xc7336 -p plcc44 -B (-B is the blank-check command). Although a blank-check automatically occurs prior to programming a device, use this stand-alone blank check if you don't know if a device has been programmed.

If all goes well, you can program the first device. Enter **xpgm** -d **xc7** 336 -pplcc44 -i **dice.prg** -P. The i option lets you to specify the input file. The **P** is the program command (it must be upper case).

FINAL CAVEAT

The only other equipment needed to start developing with Xilinx EPLDs is a good UV eraser. I frequently hear about flaky operation arising from incomplete erasure of quartz-window devices. So, here's a tip.

Xilinx EPLDs are state-of-the-art devices built with fine geometries.

```
733644 XC7336-15PC44

INVOUT1  PIN02
INVIN1   PIN03
INVIN2   PIN04
CLK      PIN05
MASTER   PIN06
INVOUTP  PIN07
COUNT    PIN08

ROLL     PIN15
S2       PIN16
S1       PIN17
S3       PIN18
S4       PIN19
S5       PIN20
S6       PIN'22
S7       PIN24
S8       PIN25
Q4       PIN26
Q1       PIN27
Q3       PIN29
02700    PIN30
Q4T00    PIN33
Q1T00    PIN34
Q2       PIN35
Q8       PIN36
Q6T00    PIN37
08700    PIN38
Q5       PIN39
Q7       PIN40
Q5T00    PIN43
Q6       PIN44
```

Figure 3—*The file DICE.VMF is used to implement pin-locking. It assigns a signal name to a specific physical EPLD pin.*

Listing 1--The file *DICE.PLD* contains the behavioral design equations *for* an electronic dice game.

```
CHIP DICE73 XEPLD

FASTCLOCK clk master
OUTPUTPIN  /q1 /q2 /q3 /q4 /q5 /q6 /q7 /q8
OUTPUTPIN  /q1too /q2too /q4too /q5too /q6too /q8too
OUTPUTPIN invoutl invout2 count
INPUTPIN   invinl invin2 s8 s7 s6 s5 s4 s3 s2 sl roll

EQUATIONS ; Logic for an RC oscilator. Roll gates the oscillator.
invoutl = /invinl
invout2 = /invin2
count = invout2 * /roll

; The counter uses q1-q3 for the right die and q4-q8 for the left
q1.clkf = clk
ql  := /q1 * q2 * q3
qltoo = q1
q2.clkf = clk
q2 := /q1  * q3 + /q1 * q4
q2too = q2
q3.clkf = clk
q3 := /q3
q4.clkf = clk
q4 := /q1 * q2 + /q1 * q4
q4too = q4
q5.clkf = clk
q5 := q1 * /q5 * q6 * q7 + /q1 * q5
q5too = q5
q6.clkf = clk
q6  := ql  * /q5 * q7 + q1 * /q5 * q8 + /q1 * q6
q6too = q6
q7.clkf = clk
q7  := q1 * /q7 + /q1 * q7
q8.clkf = clk
q8 := q1 * /q5 * q6 + ql  * /q5 * q8 + /q1 * q8
q8too = q8
```

They're more difficult to erase than a typical microcontroller or EPROM fabricated with 1.0+ micron feature sizes because there's less area per cell to absorb the UV radiation.

Xilinx has seen cases in which erratic devices pass a blank-check but are apparently not blank. After erasing for two hours and then reprogramming, the device performs properly.

As a guideline, I've found that 30 minutes is usually adequate, but your mileage may vary! Just remember that if you experience unexplainable erratic operation with a windowed device, a strong dose of UV light usually cures the problem.

Building the dice game is a good way to get started. Once you get it working properly, use your imagination to expand it. Since it requires less than half the resources of the XC7336, there's plenty of room for growth. ▣

J. Conrad Hubert is a principal in Deus Ex Machina Engineering. He provides consulting for hardware and software development of digital, analog, and mixed-signal applications. You may reach him at (612) 645-8088.

SOURCES

XPGM Programmer
Deus Ex Machina Engineering, Inc.
1377 Spencer Rd. W
St. Paul, MN 55108-5206
(612) 6458088
Fax: (612) 6450184

DS-550 EPLD **translator**
Digi-Key Corp.
701 Brooks Ave.
S. Thief River Falls, MN 56701
(218) 681-6674
Fax: (218) 681-3380

Xilinx, Inc.
2100 Logic Dr.
San Jose, CA 95 124-3400
(408) 559-7778
Fax: (408) 559-7114

I R S

401 Very Useful
402 Moderately Useful
403 Not Useful

Getting Beyond the Box With Windows 95

FEATURE ARTICLE

Craig Pataky

Interfacing gadgets to the PC was easy until Windows 95.... Although user applications can't touch hardware directly, Craig shows us how to create helper programs that access any memory location or port.

For as long as I can remember, I've been building one nonstandard gadget or another and interfacing it to my PC. Life was good until Windows 95 built a wall between the programmer and the hardware. This wall supposedly protects us, making our lives easier by releasing us from the burden of low-level mucky muck.

Okay, fine-if you're an MIS freak and happy to spend your career programming new and improved database queries. But, for intrepid souls who pick up a soldering iron now and then, the wall keeps our custom peripherals imprisoned in the land of DOS-unimpressive, unmarketable, and soon forgotten.

Yet, all is not lost.

REACHING OUT

Although user-level applications can't touch the hardware directly, you can create helper programs that interact with the hardware on the application's behalf. These helper programs are called Virtual Device Drivers (VxDs). VxDs access any memory location and any port, which means they can do all the things we took for granted under DOS.

Imagine the relationship between the application, VxD, and hardware as a three-layer pyramid. The very bottom layer represents physical hardware (e.g., parallel ports, game ports, custom devices, etc.).

The middle layer represents VxDs with the specific knowledge needed to control associated devices. The VxD also sports some form of API to communicate with the next layer up.

Finally, at the very top is the applications layer. It presents an attractive GUI to let the user do something potentially useful with the underlying mess.

To provide a concrete example of this relationship, I created an application **(LPTAPP)** and a helper VxD **(LPT CON)** that lets a user read and write the parallel port.

From the bottom up, I'll discuss the highlights of each module necessary to perform this normally off-limits operation. I won't cover every detail of Windows 95 application development and VxD writing, but the information presented here should give you an excellent start.

PARALLEL PORT

Even the most basic parallel ports have **12** readily accessible TTL outputs and 5 inputs. They are my method of choice for controlling or reading most devices that aren't equipped to handle RS-232 communications. For a pinout of useful signals, see Table **1.**

Unfortunately, outputs DBO-DB7 and CBO-CB4 are each rated to source a paltry 2 mA. If your goal is to drive a

Pin	Direction*	Signal
1	out	control bit 0 (inverted)
2	out	data bit 0
3	out	data bit 1
4	out	data bit 2
5	out	data bit 3
6	out	data bit 4
7	out	data bit 5
8	out	data bit 6
10	out	data bit 7
11	in	data bit 6
	in	data bit 7 (inverted)
12	in	data bit 5
13	in	data bit 4
14	out	control bit 1 (inverted)
15	in	data bit 3
16	out	control bit 2
17	out	control bit 3 (inverted)
18-25	gnd	—

“Direction is relative to CPU

Table I--The *25-pin parallel port is standard on all IBM-compatible PCs* and *offers* 17 pins *capable of doing useful work (12 outputs and 5 inputs).*

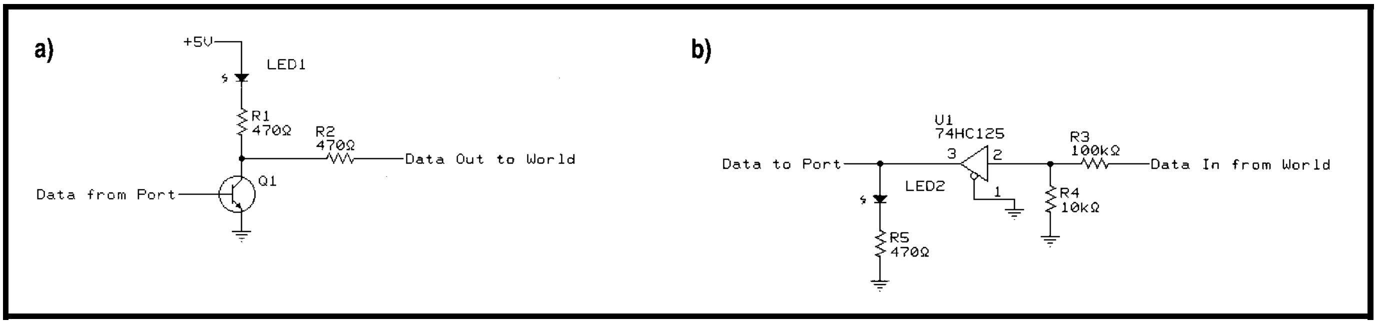


Figure 1—*(a)* **Simple** *driver and buffer circuits work well for this project. The LEDs are added to give the operator feedback. The* **output** *buffer easily drives a typical 5-V relay. You can increase V_{CC} to 9 V if your target device is far from the PC or put several of these drivers in parallel to source more current. (b) To protect your PC, use an input buffer of some kind, like the 74HC125.*

DC motor, relay, or even a single LED, you probably need to build an output driver or buffer that's up to the task. A simple transistor usually fits the bill.

Since many parallel ports are integrated into the motherboard, I strongly recommend using a buffer of some kind to protect your computer. For example, the 74HC125 is noninverting, readily available, and easy to replace when socket-mounted.

The driver and buffer circuits in Figure 1 work well in almost any application. I happen to use these circuits to drive a relay and control the temperature of an electric kiln.

LPTCON VxD

To bridge the gap between the application program and the parallel port, I created LPTCON. LPTCON is a dynamically loadable VxD that, at the request of an application, reads or writes parallel ports 1-3. Here, I'll highlight a few important areas.

In Windows 95, a 32-bit application normally requests a function from a VxD via the `DeviceIOControl` API call. By declaring a service handler for this, the VxD processes application requests easily and efficiently. Listing 1 shows LPTCON's `DeviceIOControl` handler.

As you can surmise from Listing 1, a VxD's `DeviceIOControl` handler receives just one argument which is a pointer to an IOCTLPARAMS structure. Among other things, this structure contains such information as the service being requested (e.g., `dioc_IOCtlCode`) and pointers to buffers that may contain information pertinent to that request (e.g., `dioc_OutBuf`, `dioc_InBuf`).

Listing 1—*The `DeviceIOControl` handler provides a way for 32-bit applications to request services from the LPTCON VxD.*

```
DWORD OnW32Deviceiocontrol(PIOCTLPARAMS p){
   LPTCONDATA LPData;
   switch (p->dioc_IOCtlCode){
     case LPTCON_GETVERSION: //We're being loaded
         GetBaseAddresses();
         return (0);
     case LPTCON_READ:
         //Copy the caller's LPTCONDATA structure into our own
         memcpy(&LPData, p->dioc_InBuf, sizeof(LPTCONDATA));
         //Call the function that reads the data and pass
         //it p->dioc_OutBuf which is a pointer to where the caller
         //expects us to put the byte read
         return (ReadData(LPData.LPPort, p->dioc_OutBuf));
     case LPTCON_WRITE_DATA:
         //Copy the caller's LPTCONDATA structure into our own
         memcpy(&LPData, p->dioc_InBuf, sizeof(LPTCONDATA));
         //Call the function that does the writing
         return (WriteData(LPData.LPPort, LPData.LPByte));
     case LPTCON_WRITE_CONTROL:
         memcpy(&LPData, p->dioc_InBuf, sizeof(LPTCONDATA));
         return(WriteControl(LPData. LPPort, LPData.LPByte));
     case LPTCON_CLOSEHANDLE: //We're being unloaded
         return (0);

   return (1); //Return (1) if control code unrecognized
}
```

The control code `LPTCON_GETVERSION` is asserted when the VxD is loaded into memory. I use this time to reach into the BIOS data area for each parallel adapter's base port address.

If LPTCON's `DeviceIOControl` handler determines that a read or write operation is being requested, the `ReadPort`, `WriteData`, or `WriteControl` function is called. It is passed the port number and necessary data.

The functions first retrieve the base port address for the specified port from the BIOS data area. They perform the read or write operations on that port.

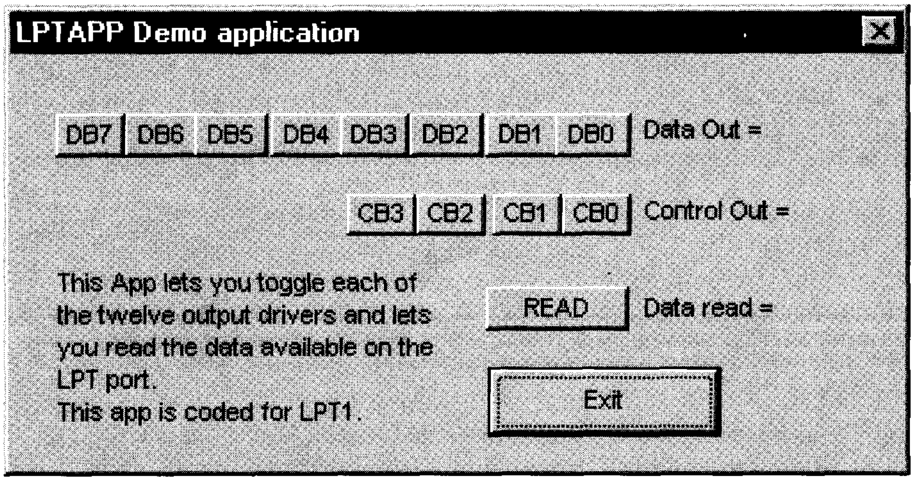
The well-equipped VxD author should have at least MASM 6.11, MSVC++ 2.0, the Microsoft DDK, NuMega's SoftICE for Windows 95 debugger, and Vireo Software's VtoolsD package. This ensemble of tools enables you to create and debug VxDs in any combination of C and assembly.

For more information on writing VxDs in general, consult the online help and samples found on the DDK and VtoolsD. (Be sure you don't have any pressing matters beforehand. It's quite engaging.]

WRAPPING LPTCON

To make it easier for an applications developer to use the LPTCON VxD, I created function wrappers that hide the gory details of the application-to-VxD interface. Instead of a series of cryptic `DeviceIOControl` calls, applications access LPTCON's services with functions that reflect the operation being performed.

I've packaged the LPTCON wrappers as static functions in LPTCON.H. If you `#define USES_LPTCON`, `#include`



This App lets you toggle each of the twelve output drivers and lets you read the data available on the LPT port.
This app is coded for LPT1.

Photo 1—*The* LPTAPP *application takes advantage of the* LPTCON *VxD to perform the MO with the parallel port.*

LPTCON.H, and ensure LPTCON.VXD is in your application's home directory, your application can access the parallel port with the following functions:

- BOOL LPTCON_LoadVxD() loads the VxD from the disk and ensures that it's ready to provide services
- BOOL LPTCON_WritePort(WORD Port, BYTE Byte) writes Byte to pins 2-9 of port Port (1-3)
- BOOL LPTCON_WriteControl (WORD Port, BYTE Byte) writes Byte to the control lines of port Port
- BOOL LPTCON_ReadPort(WORD Port, BYTE* Byte) reads a byte from port Port and puts it into Byte

Functions return true if successful or false otherwise. Concise documentation of the wrapper functions is in LPTCON.H.

LPTAPP APPLICATION

LPTAPP is a simple dialog-based application that uses LPTCON's services to allow a user to toggle each output of LPT1 by clicking on a button that corresponds to a given bit. The user can also click on the READ button to see the status of the inputs. The GUI I used for LPTAPP is shown in Photo 1.

By now, it shouldn't be mysterious how LPTAPP works internally. Each time one of buttons DB0-DB7 or CB0-CB3 is clicked, a mask value associated with the appropriate bit is XORed into the port's Data or Control image.

The LPTCON_WriteData or LPTCON_WriteControl wrapper function is called, and the wrapper function uses the DeviceIOControl API to request the write Service from the VxD. Finally, the VxD writes the image onto LPT1.

Reading LPT1 is even simpler than writing it. LPT1 is read using LPTCON_

PIC16Cxx C COMPILER

- Integrated software development environment (IDE) including an editor with interactive error detection/correction. A DOS command line compiler is also included.
- Access to all PIC hardware features from easy to use C functions.
- Built in libraries that work with all chips for RS232 serial I/O, I²C, discrete I/O and precision delays.
- Formatted Printf allows easy formatting and display in Hex or decimal.
- Source code drivers included for LCD modules, keypads, 24xx and 93xx serial E²PROM's, X10, DS1302 and NJU6355 Real Time Clocks, Dallas Touch Memory Devices, DS2223 and PCF8570 Serial SRAM, LTC1298 and PCF8591 A/D converters and more.
- Efficient function implementation allows call trees deeper than the hardware stack.
- Features such as bit variables are optimized for the unique hardware capabilities.
- Functions that call one another frequently are grouped together in the same page and calls across pages are handled automatically by the tool transparent to the user.
- Assembly code may be inserted in the source and may reference C variables.
- Integrates with MPLAB and other simulators/emulators for source level debugging. Standard Hex file and debug files ensure compatibility with all programmers.

NEW Version 2

```
#include <PIC16C71.H>
#include <stdio.h>
#use delay(clock=15000000)
#use rs232(baud=9600,
          xmit=PIN_10,rcv=PIN_7)

main() {
   int value;

   setup_port_a(ALL_ANALOG );
   setup_adc(ADC_CLOCK_INTERNAL );
   set_adc_channel( PIN_18 );

   printf("Sampling pin 18:\r\n");

   do {
     value = read_adc();

     printf("A/D value: %2x\r\n", value);
     delay_ms(1000);

   } while (TRUE);
}
```

PCB (DOS IDE) $99
for 52-58 and 12Cxx
PCM (DOS IDE) $99
for 61-84 and 14000
Professional Package $350
Both PCM & PCB with a Windows IDE
Software Prototyping Card $145
LCD/KBD for above $55
Shipping $7

CCS INC, PO Box 2452
Brookfield, WI 53008
414-781-2794 x30
http://www.execpc.com/~ccs/picc

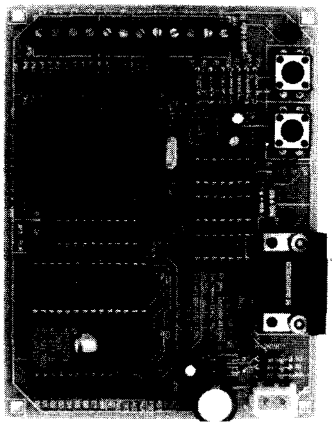
#110

8051 / C251
EMBEDDED CONTROLLERS

RIGEL Corporation introduces its newest board, the **R-51JX**, designed for Intel's **80C251** chip. RIGEL builds and supports 8 and 16 bit development tools for embedded controller systems. We feature hardware, software, books, and kits, for educational and industrial markets.

THE RIGEL DIFFERENCE
All of our boards come with:

- 32KEPROM
- 32K RAM or EEPROM
- Machine screw sockets
- Power on LED
- All system signals on headers
- I/O available on terminal blocks
- Sample programs
- Circuit diagrams
- Assembly language IDS software READS
- BBS Tech support



Our entire line of 8051 boards are programmable in Assembly, BASIC, "C", and Forth. We also offer a low cost Fuzzy Logic Software code generator and a chip simulator for the 8051 family. Complete systems start at $85
CALL TODAY FOR MORE INFORMATION

RIGEL Corporation,
PO BOX 90040, GAINESVILLE, FL 32607
Telephone (352) 373-4629 BBS (352) 377-4435

#111

Read and the resulting value is returned as shown in Listing 2.

OUT OF THE WINDOWS BOX

I've presented a bare-bones example of getting beyond the box with Windows 95. These principles apply to the serial and game ports as well.

It doesn't matter if you have relay drivers, motor controllers, temperature sensors, or robotic arms. It's time to dust off those doodads you created for DOS and give them new life as state-of-the-art Windows 95 peripherals. ❑

Craig Pataky (pronounced Puh-tah-ky) is a software engineer with Systems/Technology Development Corp. where he is involved in designing a fault-tolerant middleware layer for Windows 95. You may reach Craig at cpataky2@aol.com.

SOFTWARE

The complete source code and executables are available on the Circuit Cellar BBS.

Listing 2—*This function from LPTAPP uses a wrapper function from LPTCON.H* **to request a service** *from the LPTCON VxD.*

```
#define USES_LPTCON
#include "LPTCON.H"  //Include the wrappers
//Call the LPTCON wrapper function to ask VxD to read LPT1
int ReadPort()

  BYTE Data;
  LPTCON_Read(LPT1,&Data);
  return((int)Data);
```

SOURCES

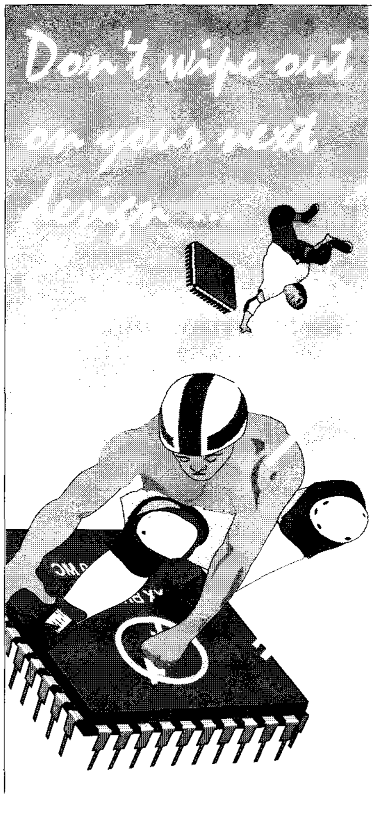
Windows 95, DDK, MASM 6.11, MSVC++ 2.0
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
http://www.microsoft.com/

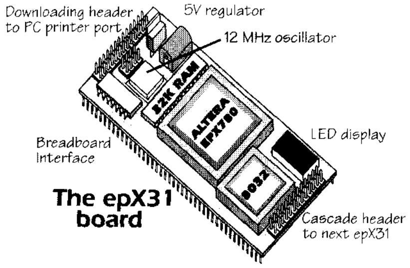
VtoolsD
Vireo Software, Inc.
21 Half Moon Hill
Acton, MA 01720
(508) 264-9200
Fax: **(SOS)** 264-9205
sales@vireo.com

SoftICE
Nu-Mega Technologies
9 Townsend
Nashua, NH 03063
(603) 889-2386
Fax: (603) 889-1135

IRS

404 Very Useful
405 Moderately Useful
406 Not Useful





FEATURE ARTICLE

David Prutchi

Designing Medical Electronic-Device Prototypes

Part 1: Design for Electrical Safety

Safety enters a new dimension when you're designing electrical devices for medical purposes. David looks at electrical safety compliance standards, using sample circuits to illustrate how to "get safe".

Military downsizing, government cutbacks, and corporate reengineering had the opposite effect on the medical industry as they did on all other areas of technology. As R&D budgets shrank, early-generation technologies considered obsolete for space, security, and military applications thrived in medical-device development.

Cruise-missile tracking technology has been adapted to steer x-ray beams so they can precisely destroy brain and spinal-cord tumors without surgery. Such dual-use technologies continue to appear with no end in sight, making medical electronics one of today's fastest growing and most promising technology-based industries.

Fortunately for entrepreneurs, prototypes of new medical instruments can be developed in a small business without esoteric technologies. A fresh idea, a personal computer, and simple interface circuitry is all it may take.

However simple or complex a medical electronic-instrument prototype may be, safety must be the primary objective throughout development. Becoming intimately familiar with electrical safety standards is critical.

The dangers involved in interfacing with the human body are often counterintuitive to an otherwise knowledgeable engineer. Did you know that a 60-Hz current of barely $10\ \mu A$ flowing through the heart can cause permanent damage and even death?

In this two-part article, I introduce the basics of designing and constructing electrically safe medical-instrument prototypes. First, I present an overview of electrical safety-compliance requirements and then proceed to look at a number of circuits that enable safe interfacing with medical electronics.

In Part 2, I'll review safety-testing methods and show the construction of several useful test instruments suitable for assessing the electrical safety of medical electronic instruments.

SHOCK PROTECTION

A long time ago, medical electronic devices left the realm of experimentation and became modern medicine's irreplaceable tools. This widespread use of electronic devices compelled countries to impose regulations ensuring their efficacy and safety.

In the U.S., the Food and Drug Administration (FDA) regulates medical devices. In the European Union (EU), a series of directives establishes the requirements that manufacturers of medical devices must meet before they obtain the CE marking that authorizes their products for sale and use.

In general, safety regulations for medical equipment address the risks of electric shock, fire, burns, or tissue damage because of contact with high-energy sources, exposure to ionizing radiation, and physical injury due to mechanical hazards.

According to the most significant technical standard, IEC-601 [1,2], the risk of electric shock exists when an operator can be exposed to a part at a voltage exceeding $25\ V_{RMS}$ or 60 VDC. An energy risk is present for circuits with residual voltages above 60 V or residual energy in excess of 2 mJ.

The device enclosure is the first barrier preventing the operator or patient from intentionally or unintentionally contacting these hazards. The enclosure must be strong enough mechanically to withstand anticipated use and misuse of the instrument. And, it must protect against fires starting within the instrument due to failures in the circuitry.

Beyond the electrical protection of the enclosure, the instrument's circuitry needs other safety barriers to keep leakage currents within allowable limits. Patient and operator safety must be ensured under both normal and single-fault conditions.

So, regulatory agencies have classified the risks posed by various parts of a medical instrument and have imposed specifications on the isolation barriers between different parts.

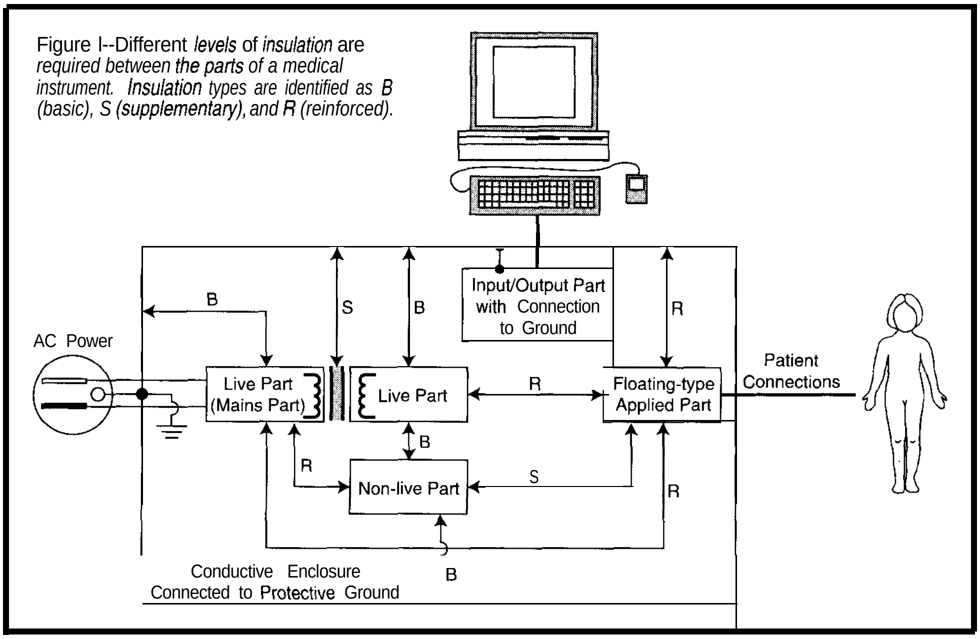


Figure I--Different *levels of insulation are required between the parts of a medical instrument. Insulation types are identified as B (basic), S (supplementary), and R (reinforced).*

The first type of part-the accessible part-can be touched without using a tool. Touching includes all contact-intentional or accidental-with the exterior of the enclosure or any exposed control knob, connector, display, or opening.

A second type is the live part. When it's contacted, current leaks beyond established limits to ground or to an accessible part of the equipment. One live part is the mains part, a circuit directly connected to the power line.

The third type, composed of signal-input and -output parts, is circuitry that interfaces the medical instrument to other instruments, which may display, record, or process data.

The fourth and most critical part of a medical instrument deliberately comes into physical contact with the patient. It includes a number of patient connections providing electrical pathways between the device and the patient. The patient circuit is composed of all patient connections and instrument parts and circuits not electrically isolated from these connections.

Levels of electrical-shock protection provided to the patient by the isolation of applied parts are classified as:

- Type B-applied parts providing a direct ground connection to the patient
- Type BF-applied parts "floating" (hence the F). The applied part is isolated from all other parts of the equipment to such a degree that the leakage current flowing through the patient to ground does not exceed the allowable level even when a voltage equal to 110% of the rated AC voltage is directly applied between the applied part and ground.
- Type CF-applied parts, similar to type BF, but providing a higher degree of protection enabling direct connection to the heart

Type-F applied parts are preferred in all cases to Type-B applied parts since patient environments often involve the simultaneous use of multiple electronic instruments connected to the patient. Type-B applied parts are prohibited whenever patient connections provide either low-impedance or semipermanent connections to the patient (e.g., through recording bioelectrodes in ECG or EEG).

AC Voltage	DC Voltage	Basic Insulation		Double or Supplementary Insulation		Double or Reinforced Insulation	
		Air Clearance	Creepage Distance	Air Clearance	Creepage Distance	Air Clearance	Creepage Distance
125	150	1	2	1.6	3	3	6
250	300	1.6	3	2.5	4	0.25	[illegible]
300	450	2.4	4	3.5	6	7	12

Table **1--The** *standards require various spacings (in millimeters) to provide different levels of insulation between* parts *of a medical instrument.*

Medical electrical equipment intended for direct cardiac application (e.g., intracardiac electrophysiology catheters) must contain only Type-CF applied parts. In addition, the applied parts of instruments for cardiac diagnosis and therapy often withstand the application of high-voltage, high-energy shocks like those used for cardiac cardioversion and defibrillation [3].

These classifications are more than academic. The standards clearly indicate the minimal level of circuit separation and acceptable levels of insulation between parts. Insulation is not only solid insulating material applied to a circuit, but also to spacings which establish creepage distances and air clearance between parts.

The separation of two conductive parts by air alone constitutes a clearing distance, while the separation of conductive parts on a nonconductive plane (e.g., traces on a PCB) is a creepage distance. The minimum separation distance between two parts is determined by the working voltage between the parts and by the insulation rating required to provide protection against electrical shock.

An insulation barrier applied to live parts provides basic protection against electrical shock. It separates a live part and an accessible conductive part that is protected by connection to ground.

Supplementary insulation is an independent insulation barrier applied in addition to basic insulation. It protects against electrical shock if basic insulation fails. Double and reinforced insulations provide protection equivalent to both basic and supplementary insulations.

Figure 1 and Table 1 present a partial view of how to achieve the minimal required insulation ratings between parts. Although these are only a subset of all possibilities contemplated by the standards, they provide a practical reference for the designer.

Equipment Type	Type B		Type BF		Type CF	
Condition	Normal	Single Fault	Normal	Single Fault	Normal	Single Fault
Ground Leakage Current	0.5	1	0.5	1	0.5	1
Enclosure Leakage Current	0.1	0.5	0.1	0.5	0.1	0.5
Patient Leakage Current	0.1	0.5	0.1	0.5	0.01	0.05
Patient Leakage Current (with powerline voltage on the Applied Part)	–	–	–	5	–	0.05
DC Patient Auxiliary Current	0.01	0.05	0.01	0.05	0.01	0.05
AC Patient Auxiliary Current	0.1	0.5	0.1	0.5	0.01	0.05

Table 2—*These are some of the maximum values of continuous leakage and* **patient** *auxiliary currents (in milliamperes) allowed by UL Standard 2601-1[4].*

LEAKAGE CURRENTS

Isolation barriers ensure that leakage currents are maintained within safe values even when a single-fault condition occurs. Three types of leakage currents are defined within the standards:

- Ground-current flowing from all AC parts through or across the insulation into the protective ground conductor of the grounded power cord
- Enclosure-total current flowing from the enclosure and all accessible parts (excluding applied parts) through an external conductive connection other than the protective ground conductor to ground or another part of the enclosure
- Patient-current flowing from the applied part via the patient to ground or from the patient via an F-type applied part to ground. These currents originate from the unintended appearance of a voltage from an external source on the patient.

These leakage currents must not be confused with currents intentionally generated by the medical device to produce a physiological effect on the patient or used by the applied part to facilitate measurement without producing a physiological effect.

In the standards, the terms "voltage" and "current" refer to the root-mean-square (RMS) values of an alternating, direct, or composite voltage or current. By definition, the RMS value of an alternating voltage V across a resistor R equals the direct voltage causing the same dissipation level in R. For a sinusoidal waveform, V_{RMS} is related to the peak-to-peak voltage (V_{pp}) by:

$$V_{pp} = V_{RMS}\sqrt{2} \approx 1.414\,V_{RMS}$$

A corresponding definition applies to the value of an RMS current. In the case of composite-(AC + DC) signals, the RMS value is calculated from:

$$V_{RMS} = \sqrt{V_{DC}^2 + V_{AC_{RMS}}^2}$$

As shown in Table 2, allowable patient leakage and auxiliary currents are defined for both normal and single-fault conditions. This table assumes that the equipment is operating at maximum load and that the supply is set at 110% of the maximum rated supply voltage.

In single-fault conditions, the single means of protection against a safety hazard in the equipment is defective or a single external abnormal condition is present. These conditions include interruption of the supply by opening the neutral conductor and the interruption of the protective ground conductor.

Patient leakage current between a Type-F applied part and ground assumes that an external voltage equal

DISCLAIMER

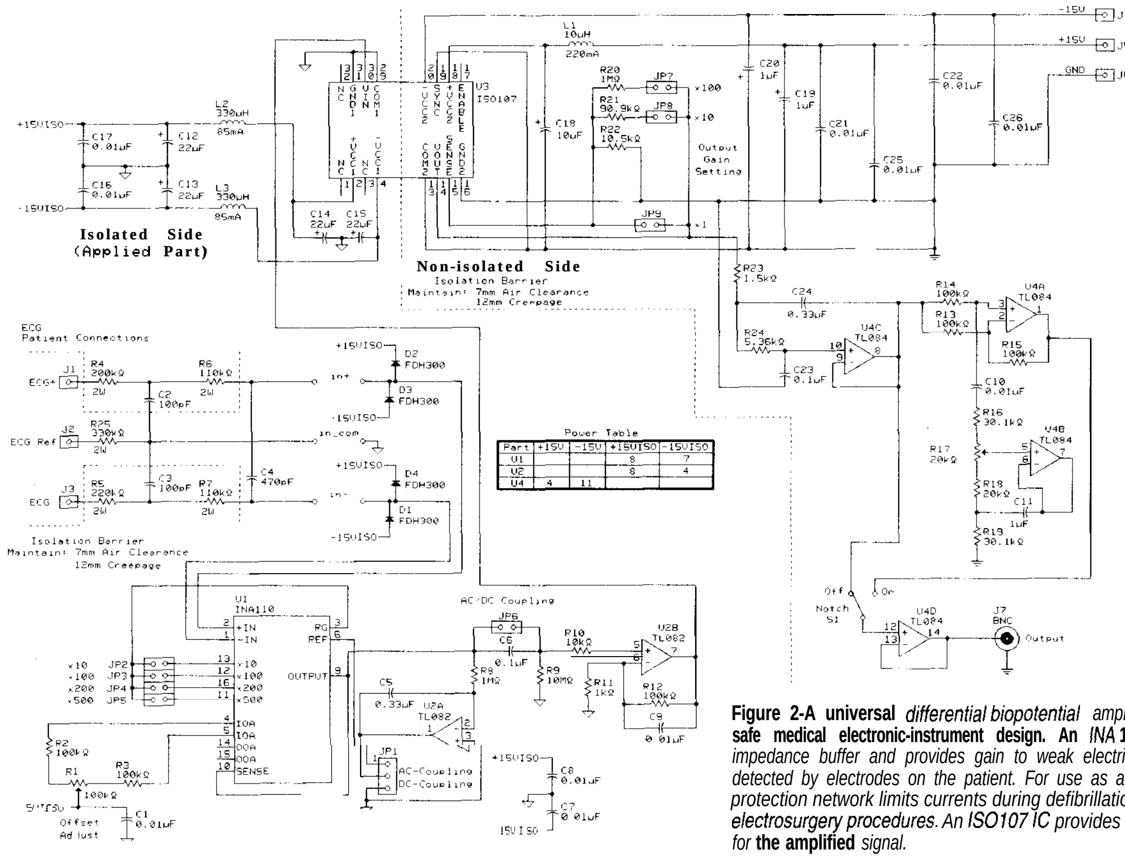
The circuits in this article are examples of engineering building blocks used in the design of experimental bioelectronic instruments. References to various safety standards are made for illustrative purposes only and are not intended to be used instead of the complete standards or as a guide to standards for a particular application.

The author does not suggest that the equipment, methods, or circuitry presented herein can or should be used by the reader or anyone else to conduct experiments with human subjects or experimental animals. Neither does the author suggest that they be used in place of or as an adjunct to professional medical treatment or advice.

Please note that the risks of electrocution and fire exist for some of the circuits presented herein. Sole responsibility for the construction and use of these circuits or of systems incorporating these circuits lies with the reader, who must apply for any and all approvals and certifications that the law may require for their construction, sale, or use.

The author does not make any representations as to the completeness or the accuracy of the information contained herein and disclaims any liability for damages or injuries, whether caused by or arising from lack of completeness, inaccuracy of information, misinterpretation of the directions, misapplication of the circuits and information, or otherwise.

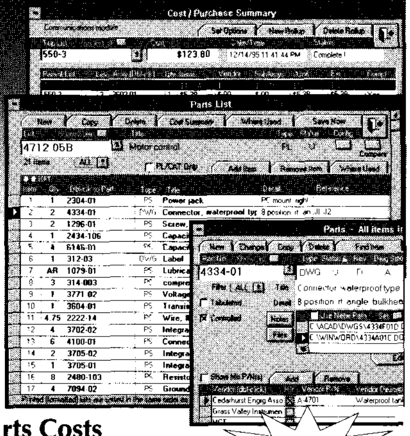
Finally, the author is not responsible to the reader or third parties for any claim of special or consequential damages, in accordance with the previous disclaimer.



Power Table

Part	+15V	-15V	+15VISO	-15VISO
U1			8	7
U2			8	4
U4	4	11		

Figure 2-A universal *differential biopotential amplifier exemplifies* **safe medical electronic-instrument design. An** *INA110* **acts as an** *impedance buffer and provides gain to weak electrical activity detected by electrodes on the patient. For use as an ECG amplifier, a protection network limits currents during defibrillation or electrosurgery procedures. An ISO107 IC provides galvanic isolation for* **the amplified** *signal.*



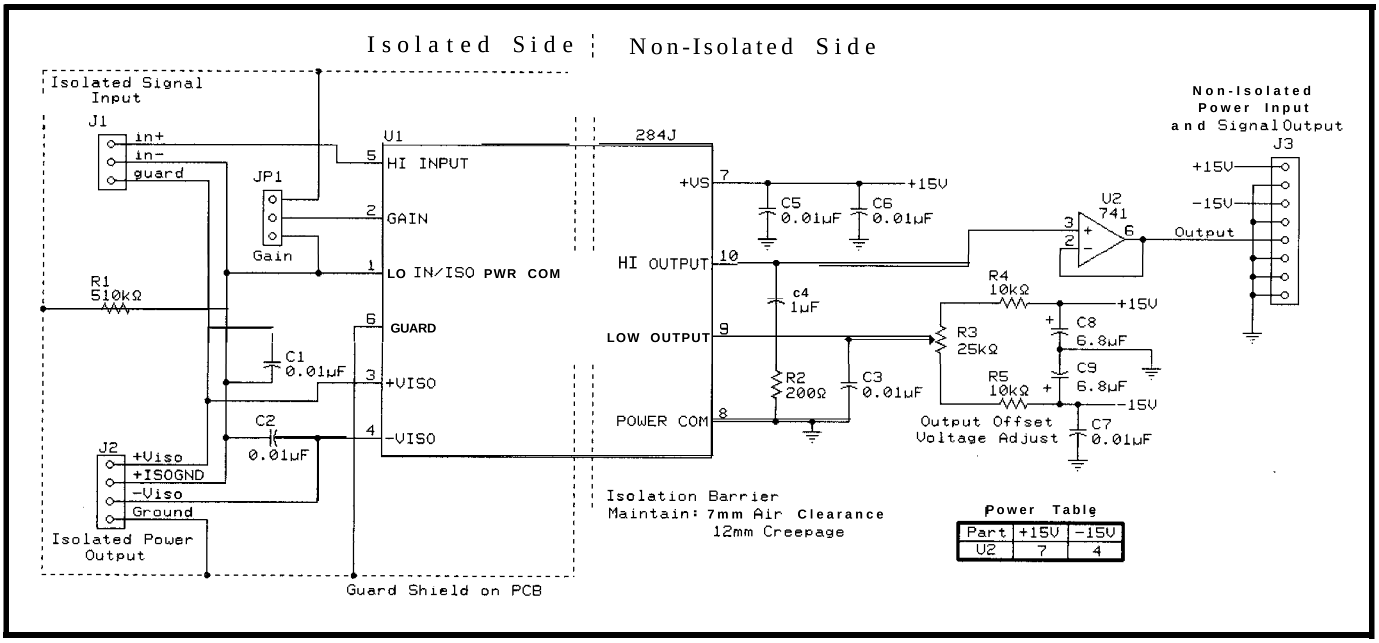


Figure 3-*A stand-alone signal isolator can be built using an Analog Devices 284J isolation amplifier. This isolation amplifier meets stringent leakage standards of 2.0 μA maximum at 115 VAC, 60 Hz, providing a maximum isolation of 2.500-V_{RMS} AC at 60 Hz for 1 min., and ±2500-V_{p-p} maximum continuous AC, DC, or 10-ms pulses at 0.1 Hz*

to 110% of the maximum rated supply voltage is directly connected to the applied part. For battery-powered equipment, the external voltage assumed to be connected to the Type-F applied part is 250 V. (The grounding of the patient is considered normal.)

These allowable leakage current levels are a compromise between achievable performance and overall risk. Although a 60-Hz current as low as 10 μA flowing through the heart may cause ventricular fibrillation (a disorganized quivering of the lower chambers of the heart muscle which quickly leads to death) under highly specific conditions, the probability of this event is only 0.2%.

Under more realistic clinical conditions, a 50-μA current flowing from a Type-CF applied part through an intracardiac catheter has an overall probability of 0.1% of causing ventricular fibrillation. This probability is close to that of causing fibrillation by mere mechanical contact of the catheter with the heart wall.

Obviously, for equipment that does not come in direct contact with the heart, allowable leakage currents have been increased. Even though the patient may perceive the actual current, it has been increased up to the point where even under single-fault condition, the probability of causing ventricular fibrillation is no higher than 0.1%.

ISOLATED BIOPOTENTIAL AMPLIFIER

Let's use a simple circuit to illustrate the various considerations for safe design of a medical instrument. Figure 2 presents the schematic diagram of a biopotential amplifier-a circuit designed to amplify and process electrical signals of biological origin.

In the circuit, signals picked up by electrodes attached to the patient's skin are amplified by U1, a Burr-Brown INA1 10 instrumentation amplifier IC. The gain of the front-end stage is programmable between unity and 500 by jumpers JP2–JP5.

Potentiometer R1 trims the input offset to U1. Rl-R3 can be omitted from the circuit for most applications not requiring extreme DC precision.

Direct connection of Ul's inputs to the patient electrodes is possible since the amplifier uses a bias current of 50 pA max. Also, the FDH300 low-leakage diodes protecting the inputs of U1 contribute no more than an additional 1nA each to the patient auxiliary current.

The total 54-nA maximum is well under the allowed 0.01-mA auxiliary current for Type-CF applied parts. If the application permits it (e.g., if the skin-electrode interface has sufficiently low impedance), it's a good idea to add resistors larger than 300 kΩ in series with the patient connections.

These resistors effectively limit the auxiliary current flowing through the patient to less than 0.05 mA in case a fault in U1 or Dl-D4 short-circuit the patient connection with one of the isolated power rails.

Depending on the biopotential signal being amplified, either DC- or AC-coupling is required. For DC-coupling, U1 is referenced to the isolated ground plane IG1, which also serves as the patient common input.

Since the INA1 10 has FET inputs, bias currents drawn through input source resistances have negligible effect on DC accuracy. However, a return path must be provided to prevent charging of stray capacitances which may saturate the INA1 10. Whenever floating sources or capacitively coupled sensors are used, a 10-MΩ resistor to the isolated ground plane from each input should be used.

When AC coupling is desired, U4a together with R8 and C5 offset Ul's reference to suppress a baseline composed of components in the DC to 0.48-Hz range. Also for AC coupling, any remaining baseline at Ul's output may be eliminated by a high-pass filter (1.59 Hz at -3 dB) formed by C6 and R9.

U1's output signal is amplified by U4b. Notice that the gain of this stage is fixed at 101. Galvanic isolation is provided by U3, a Burr-Brown ISO107 isolation amplifier IC.

This type of IC resembles an operational amplifier, but it's designed with an internal isolation barrier between its input and output pins. The ISO-107's signal channel has a small-signal bandwidth of 20 kHz and provides an isolation barrier rated at a continuous 2500 v.

In addition to providing a signal channel across the isolation barrier, the ISO107 has an internal DC-DC converter which powers the isolated side of the ISO107 circuitry and provides isolated power (±15 V at ±15 mA typical) for the rest of the applied part's circuitry (i.e., U1 and U2).

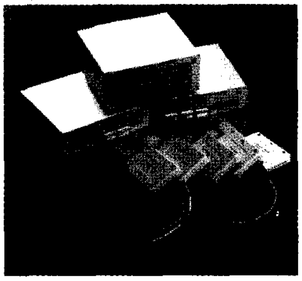
The isolation rating of the barrier for the DC-DC converter is the same as that for the signal channel. In total, the 60-Hz leakage current through U3 doesn't exceed 2 µA with 240 V applied across its isolation barrier.

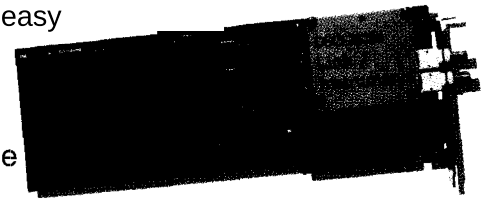
U3's output gain is selected through jumpers JP7–JP9 to provide gains of 1, 10, or 100. U3's output is then low-pass filtered by U4c. With the component values shown, the filter has a cutoff frequency of 300 Hz. Recalculate these values to match the bandwidth required by your application.

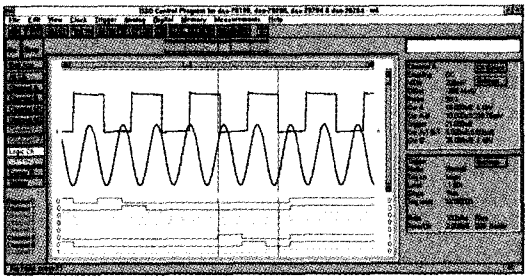
In one position of S1, the filter's output is directly buffered by U4d and presented to the output of the biopotential amplifier. In the other position, S1 redirects U4c's output to a tuneable-frequency notch filter before being buffered by U4d. This feature makes it easy to eliminate a 50–60-Hz power-line hum picked up through common-mode imbalances between the differential patient connections.

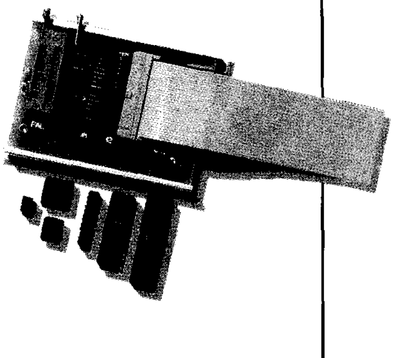
Up to this point, patient leakage and auxiliary currents are kept within allowed limits by selecting the circuit components. However, appropriate layout and interconnection is as important in ensuring a safe design.

To do so, every single conductive point belonging to the circuit's isolated portion must be separated from every conductive point in the nonisolated side of the circuit by the required air clearance and the creepage dis-









tances corresponding to reinforced insulation at the rated working voltage.

Since 30 mm separates the closest pins across the IS0107 isolation barrier and since the internal isolation barrier is rated at a continuous 2500 V at 60 Hz, the standards consider this barrier to be equivalent to 1000 VAC-rated reinforced insulation. Separation is also needed between all isolated and nonisolated points of the circuit.

Most commonly, a biopotential amplifier operates in environments where the power-line voltage is the highest potential of concern. It has a maximum rated value of 240 V_{RMS}.

According to Table 1, this requires an air clearance of 5 mm and a creepage distance of 8 mm. These distances also apply to the separation of any point in the isolated side and any conductive fastening means in connection with any nonisolated part of the medical instrument.

Amplifying the electrical activity produced by the heart introduces a number of additional requirements addressed by the front-end protection circuit shown in Figure 2. Physicians conducting electrophysiological diagnosis and therapy of conditions involving the heart assume the possibility of ventricular fibrillation during a procedure.

Reverting fibrillation back into a normal rhythm driven by the sinus node of the heart involves briefly forcing high current through the heart. To overcome tissue resistivity, a high-energy, high-voltage pulse is delivered.

Typical external defibrillators discharge a 32-µF capacitor charged up to 5000 VDC through a 500-µH inductor directly into paddle electrodes placed on the patient's chest. The body provides a 100-Ω resistance. A sizeable fraction of the defibrillation pulse may appear at the ECG recording electrodes and between the isolated patient ground and the power-line ground.

The front-end protection circuit places 330-kΩ resistors (R4, R6, R25, R5, and R7) in series with the patient leads to limit the peak defibrillator input current to under 10 mA. To withstand the several dozen watts of instantaneous power potentially dissipated during each defibrillation pulse, 2-W carbon-composition, high-voltage-rating resistors are chosen.

Voltages close to the full 5000-V defibrillator-capacitor initial voltage could appear across these resistors. Take care to ensure that current does not find an alternate path by producing a spark or creeping across the printed circuit.

The insulation required to withstand the peak voltage of the defibrillator pulse should be a minimum of 7-mm air clearance and a 12-mm creepage distance. This separation also applies to the isolation barrier between the applied part and all other instrument parts.

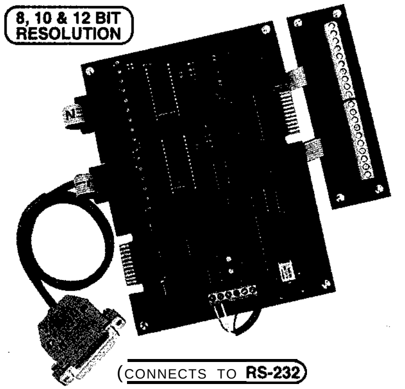
A second consideration is necessary for operating-room equipment. Here, the applied part of the instrument may be exposed to strong RF currents coming from an electrosurgery (ESU) unit used for either cauterizing wounds or cutting tissue.

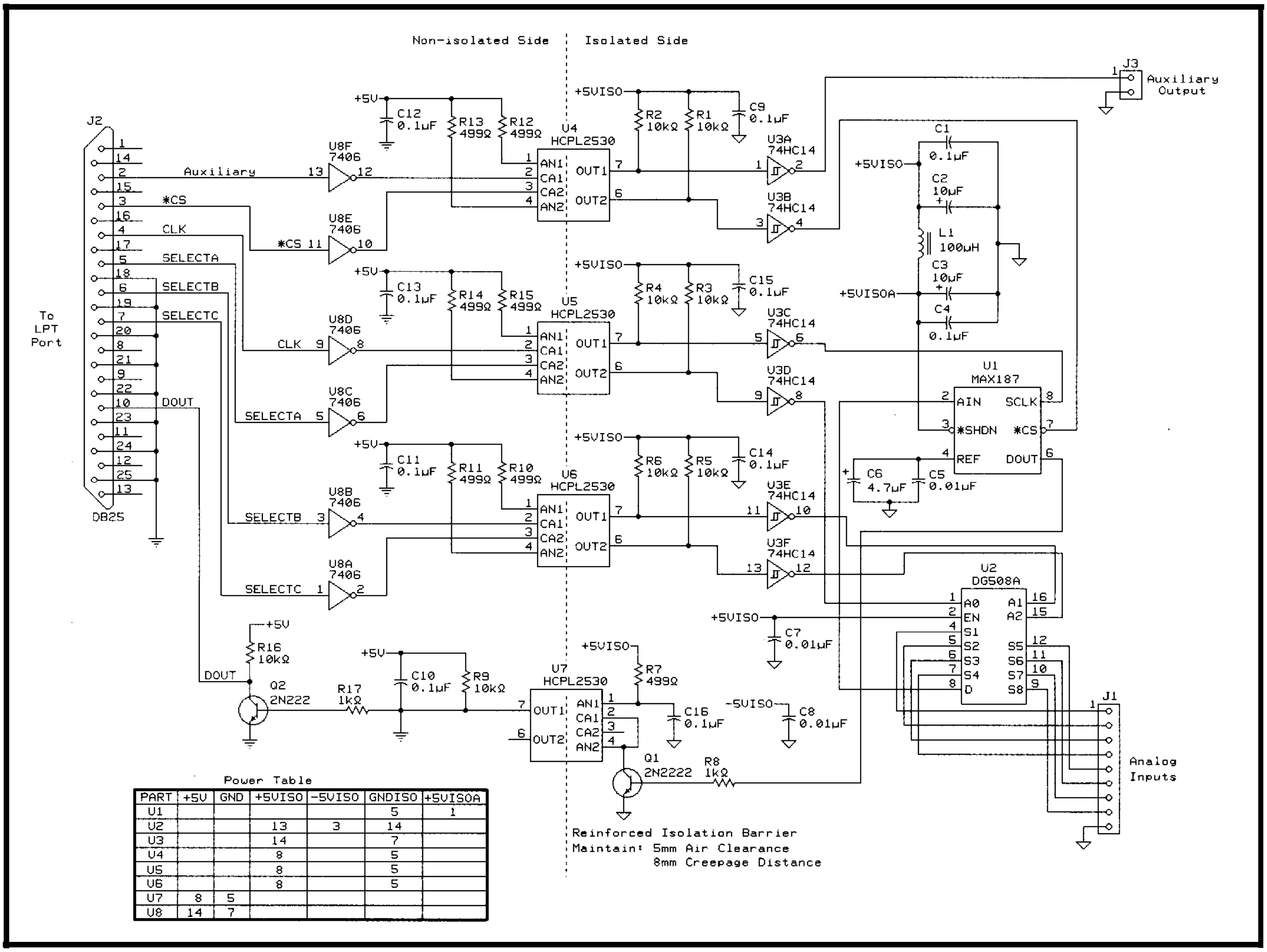
Usually, CW or gated damped sinusoids are applied between a large-area electrode on the patient's back and the scalpel electrode. Through RF heating, tissues are cut and blood is coagulated, causing small ruptured vessels to close.

The RF component of the ESU waveform ranges from 200 kHz to 3 MHz. Power levels into a 500-Ω load range from 80 to 750 W. Open-circuit voltages range from 300 V, but can be as high as 9 kV.

If the circuit of Figure 1 were used in the presence of ESU, the path of RF leakage current would probably be from the ESU electrodes, into one or more of the patient electrodes, through the IS0107 coupling capacitance, through the stray capacitance of the power supply transformer, into the power-line, and back through the stray capacitance of the ESU generator's power transformer.

To deal with these RF currents, medical electronic equipment with filters attenuate RF signals before they can be detected by the circuit's nonlinearities. In our front-end protection circuit, RF appearing at the ECG+ and ECG- electrodes is sunk to the isolated ground by C2 and C3. C4 eliminates any remaining RF differentially





Power Table

PART	+5V	GND	+5VISO	-5VISO	GNDISO	+5VISOA
U1					5	1
U2			13	3	14	
U3			14		7	
U4			8		5	
U5			8		5	
U6			8		5	
U7	8	5				
U8	14	7				

Figure 4-A *MAX187 low-power 12-bit A/D converter IC forms the core of the isolated A/D converter module. A two-wire serial interface conveys data* **from** *the* **MAX187 through** *optoisolators back to the PC* **through** *the printer port. The PC a/so controls* **an isolated** *signal multiplexer which allows one of eight analog signals to be presented to the input of the ADC.*

demodulated by Dl-D4 or Ul's circuitry.

Here again, currents driven by high RF voltages must not find alternate paths such as corona discharge or creepage. For these reasons, appropriate spacings must be observed.

STAND-ALONE ANALOG ISOLATORS

This example demonstrated the use of the circuit-embedded IS0107 isolation amplifier to provide a signal path across the isolation barrier. In many cases, a self-contained general-purpose isolation module simplifies prototype and experimental equipment design.

Figure 3 shows the circuit diagram for such a module. This module, a reliable isolation board, protects subjects connected to isolated biosignal amplifiers from lethal ground-fault currents and those originating from defibrillator pulses.

The heart of the module is U1, an Analog Devices 284J high-performance isolation amplifier. This isolation amplifier meets stringent leakage standards of 2.0 µA maximum at 115 VAC and 60 Hz.

Its remarkable performance results from the carrier isolation technique which transfers signals and power across the isolation barrier. It provides a maximum isolation of 2500 V_{RMS} AC at 60 Hz for 1 min., and ±2500 $V_{p\text{-}p}$ maximum continuous AC, DC, or 10-ms pulses at ⅒ Hz.

Bipolar input signals-present at Jl-3 and referenced to isolated ground at Jl-2-are introduced into the isolated signal input of U1. Ul's gain can be set between l-10 by changing the value of a resistor R_i connected between JP1-1 and JP1-2 according to:

$$\text{Gain} = 1 + \frac{100\,\text{k}\Omega}{10.7\,\text{k}\Omega + R_i\,[\,\text{k}\Omega\,]}$$

To preserve high CMR, this resistor and all connections to it must be guarded with a shield connected to JP1-3.

Best performance is achieved by placing a shorting jumper between JP1-1 and JP1-2 and operating the circuit at a gain of 10. Leaving JP1 open results in unity gain.

The bandwidth supported by the 284J is DC to 1kHz (small signal), DC to 700 Hz (full power G = 1), and DC to 200 Hz (full power G = 10). Ul's output is buffered by U2, a unity gain buffer, to drive low-impedance loads connected between the module's output at J3-5 and nonisolated ground at J3-6.

Trimmer R3 zeros the output offset voltage over the gain range. Ul's output is low-pass filtered to roll off noise and output ripple. Cutoff (-3 dB) of the low-pass filter is given by:

$$f_{(-3\,db)}[\mathrm{Hz}] = \frac{1}{2\pi C4[\mathrm{F}] \times 1000[\Omega]}$$

Use of a 1-µF capacitor results in a cutoff frequency of approximately 160 Hz.

This module's input voltage range is ±5 V differential at unity gain. However, this module can also be used for the direct low-level amplification of biosignals with a low-input noise (10 μV_{p-p}), medium-input impedance (10^8 R), and high CMR (110-dB inputs to output, 78-dB inputs to guard).

Differential measurement of a biopotential is achieved between J1-3 (the noninverting input) and J1-2 (the inverting input). CMR is optimized by connecting J1-1 to a distant reference electrode. By using the module as a bioamplifier, the leads to the bioelectrodes should be low-loss, low-capacitance coaxial cables, whose shields are connected to J1-1.

This module should be operated using a symmetric ±15-V regulated power supply (J3-1 = +15 V, J3-2 = nonisolated ground, J3-3 = -15 V). Dual ±8.5 VDC at 5 mA of isolated power are provided at J2. These lines power floating-input circuitry such as biosignal buffer preamplifiers.

High performance means high price, and the IS0107 and 284J are no exception. The unit price for each is over $100, making their use prohibitive in low-cost designs and in instruments that involve a large number of analog signals crossing the isolation barrier.

In these cases, analog isolators can be built by using low-cost optoisolators as isolation channels. The problem is that optoisolators have a narrow linear range. So, it's difficult to directly drive them with a full-range analog signal.

However, simple circuits which first convert the analog signal into a pulse train of variable frequency (or pulse width) can drive an optoisolator. At the other side of the optoisolator, the pulse train is demodulated to render the original signal.

Another possibility is to place the optoisolator within a servo loop which makes use of the loop's error to convey a high-linearity analog signal [5]. You could also convey true-digital data through the optoisolator. Keep on reading..

A DIGITAL ALTERNATIVE

Many modern medical electronic instruments use an embedded microcomputer or an external PC for control, data processing, and display. In most cases, an analog-to-digital converter at some point in the instrument supports data-acquisition functions.

The circuit in Figure 4 directly connects the A/D converter with the medical instrument's applied part. It relays digital rather than analog signals across the isolation barrier.

This alternative to analog-signal isolation avoids the additional noise, nonlinearity, and complexity of analog by translating signals to digital format early on within the instrument. Optoisolators for high-speed digital signals are inexpensive and widely available. In addition, serial data formats minimize the number of digital signals concurrently communicated through the isolation barrier.

Many modern high-end medical instruments make extensive use of this philosophy. If you have the opportunity, examine the circuit schematics of one of today's electrocardiography instruments. You'll probably find an elegant design comprising instrumentation amplifiers for each lead followed directly by an A/D converter and optical isolation leading to a DSP microprocessor.

Often, the complete applied part is contained within a medallion directly connected to the patient leads. Digital signals to and from the embedded microcomputer are relayed through optical fiber.

The sample circuit here is not as complex as those of high-end commercial instruments, but it provides a simple and convenient interface between analog-output applied parts and almost every PC on the market. Instead of connecting to the computer's expansion bus, it plugs into one of the parallel printer ports, which is used as a serial I/O for an 8-channel A/D converter.

At the heart of the circuit is a MAX187. This IC is one of Maxim's single-chip A/D converters featuring a 12-bit 8.5-µs successive-approximation converter, a 1.5-µs track-and-hold, on-chip clock, a precision 4.096-V reference, f0.5 LSB nonlinearity, and a high-speed three-wire serial interface.

One of eight isolated analog signals to be measured is presented to the analog input line AIN of the MAX187 via a DG508A analog multiplexer IC. Voltages between 0 and 4.096 V are converted by the ADC into distinct digital codes for every 1 mV of change.

MAX187's A/D conversion initiation and data-read operations are controlled by the *CS and SCLK (Serial Clock) lines.

As shown in Figure 5, A/D conversion is initiated by a falling edge on the *CS line. At this point, the track-and-hold holds the input voltage and

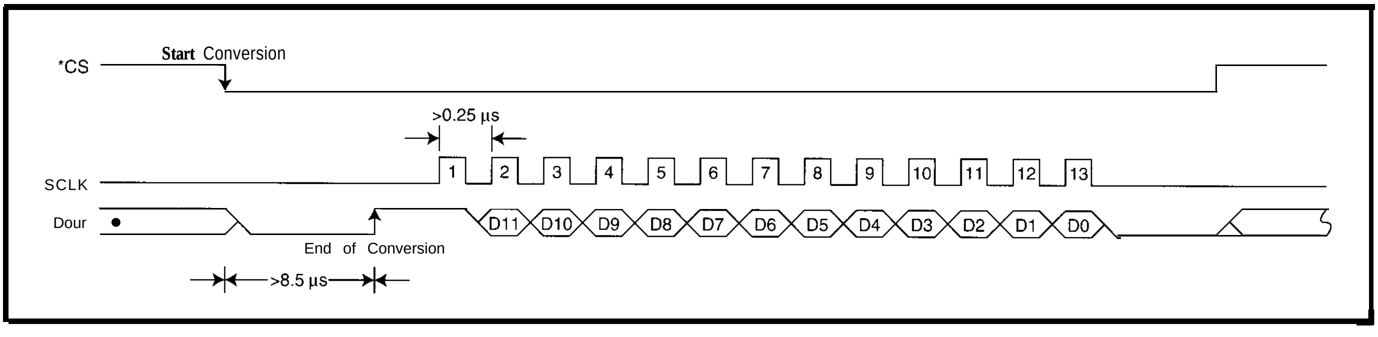


Figure 5—*This graph shows data acquisition and serial protocol timing for the MAX187. AID conversion is initiated by a falling edge on the ● CS line. After conversion, data is read out in serial format, shifted from the sequential-approximation* **register** *on* **each falling-edge transition of** *SCLK.*

Listing 1--This *QuickBasic* sample *program acquires data using the B-channel, isolated 12-bit A/D converter. The use of LPT1: is assumed.*

```
' Printer port locations
CONST prinop = &H378   ' Printer Output Port (&H278 or &H3BC)
CONST prinstat = &H379 ' Printer Status Port (&H279 or &H3BD)
'
' Define control pin locations
CONST aux=1, notcs=2, sclk=4, sela=16, selb=32, selc=64

' Initialize
OUT prinop, 0                    ' clear printer por
CLS                              ' clear screen
INPUT "Please input channel to acquire "; chan
' determine control bits for desired channel
selas = (chan AND 1) * sela
selbs = (chan AND 2) / 2 * selb
selcs = (chan AND 4) / 4 * selc
'
' Acquisition and display control
start:
SCREEN 2                         ' CGA graphics mode 640x200
GOSUB acquire                    ' determine first d splay point
y = INT((4.096 vout) * 45) + 10' compute pos of start point
startl:
CLS                              ' refresh screen
LOCATE 2, 2: PRINT "4.096V";     ' place y-axis labels
LOCATE 7, 2: PRINT "3.000V";
LOCATE 13, 2: PRINT "2.000V";
LOCATE 19, 2: PRINT "1.000V";
LOCATE 25., 2: PRINT "0.000V":
PSET (60, y)                     ' place first sample
FOR x = 60 TO 640                ' horizontal sweep
  GOSUB acquire                  ' acquire a sample
  y = INT((4.096 vout) * 45) + 10' compute position on screen
  LINE -(x, y)                   ' display data
   IF INKEY$ <> "" THEN GOT0 progend' press any key to escape
NEXT x                           ' next sample
GOT0 start1                      ' start a new screen
progend:
'
' Leave program
OUT prinop, 0                    ' clear printer port
SCREEN 0                         ' return to text mode screen
END
acquire:

' Acquisition loop
OUT prinop, selas + selbs + selcs + notcs' keep CS' deasserted
convert:
OUT prinop, selas + selbs + selcs' convert by asserting CS'
loop1:
bit = (INP(prinstat) AND 64) / 64' read stat port & filter DOUT
 IF bit = 0 THEN GOT0 loop1      ' wait for EOC signal
dat = 0                          ' clear A/D accumulator
FOR clocknum = 11 TO 0 STEP -1' clock 12 bits serially
  OUT prinop, selas+selbs+selcs+sclk' clock pulse rising edge
  OUT prinop, selas+selbs+selcs     ' clock pulse falling edge
  bit = (INP(prinstat) AND 64) / 64 ' read stat port & filter DOUT
  dat = dat + (2 ^ clocknum) * bit  'acc from bit 11 to bit 0
NEXT clocknum                    ' next bit
OUT prinop, selas + selbs + selcs + sclk ' clock to reset A/D
OUT prinop, selas + selbs + selcs
OUT prinop, selas + selbs + selcs + notcs' deassert CS'
vout = dat * 0.001               ' translate A/D data to volts
RETURN
```

the successive-approximation process commences.

The start of conversion is acknowledged by the MAX1 8 7 changing the state of the DOUT line from high to low impedance. After an internally timed 8.5-µs conversion period, the end of conversion (EOC) is signaled by the DOUT line going high.

Once a conversion completes, data can be obtained in serial format shifted from the sequential-approximation register on each falling-edge transition of SCLK. Since there are 12 bits, a minimum of 13 falling-edge pulses are required to shift out the ADC result.

U4–7 provides the isolation between the PC's printer port and the

MAX187. Bits 1 and 2 of the PC's LPT 8-bit output port (hex address 378 for LPT1:) are toggled by software to implement the control portion of the MAX187 serial protocol.

Bit 6 of the printer-status port register (hex address 379 for LPT1:) receives the serial data from the MAX187. Bits 3-5 of the output port control the analog signal multiplexer. Bit 0 of the output port is an auxiliary line which can control the applied part's circuitry.

The MAX1 87 is powered from a patient-contact-rated isolated power supply capable of delivering ±5 V. A π filter formed by Cl-4 and L1 ensures a clean supply to the ADC.

Notice that two separate isolated ground planes, one analog and one digital, are shown in Figure 4. Ideally, the signal ground plane is used as the reference for the analog input signal.

This same ground plane should be constructed to shield the analog portions of the ADC and signal multiplexer (i.e., the input-network and the voltage-reference filtering and decoupling capacitors). The analog and digital ground planes should be connected at a single point, preferably directly to the isolated ground line supplying the circuit.

SOFTWARE FOR THE ISOLATED ADC

Listing 1 presents a sample program for driving the isolated A/D converter from a PC printer port. The program flow starts by initializing the ports. The use of the standard LPT1: is assumed. You may need to change the output- and status-port locations to suit your specific installation.

This example program runs the A/D converter as fast as the PC is able to drive its lines. Sampling rate control could be implemented by either adaptively inserting FOR NEXT loops to introduce controlled delay between samples-see "LPT:*Analog!*" (*INK* 67)-or by controlling the acquisition process from interrupts generated by high-resolution hardware timing (see D.P. Schulze's "A PC Stopwatch" (*INK* 19) and B. Ackerman's "High-Resolution Timing on a PC" (*INK* 24)).

The acquisition subroutine sets up the multiplexer while keeping ● CS deasserted. Conversion for the selected channel is initiated by asserting ● CS and polling for the EOC signal before attempting to read the conversion data.

At this point, the ADC accumulator variable is cleared, and each of the 12 bits are clocked in serially. The value of each bit is read from the status port and is multiplied by the decimal value of its binary position before being accumulated.

Finally, one more clock pulse resets the ADC, the ● CS line is deasserted, and ADC data is translated to volts.

POWER SUPPLIES

As Figure 1 showed, reinforced insulation between the applied part and other instrument parts doesn't mean that similarly strong insulation is not needed between an AC part and other live or nonlive parts besides the applied part.

Although a component may in itself power the applied part across an appropriate isolation barrier (e.g., ISO107, 284J, etc.), the instrument's power supply must still meet the same requirements as a safety isolation transformer.

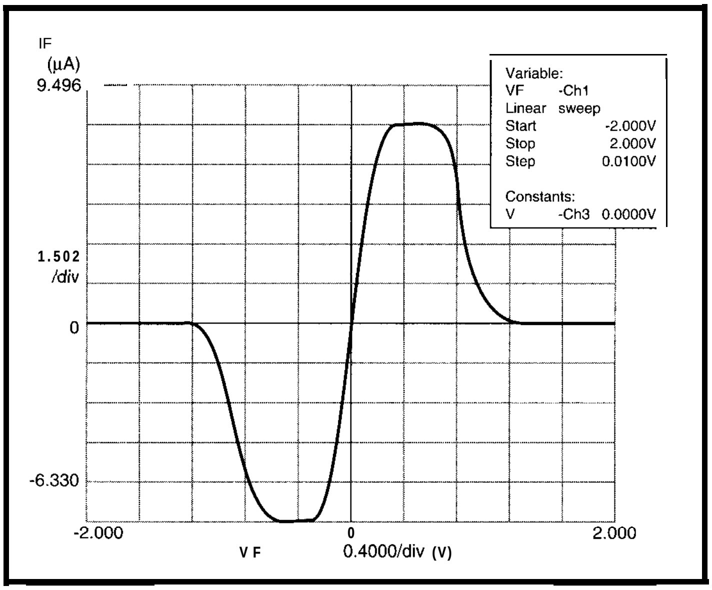
To accomplish the required levels of isolation, medical instruments often incorporate a safety extra-low-voltage (SELV) transformer to derive their operating power from the power line. This transformer type supplies a voltage under 25 VAC or 60 VDC.

It has an output winding which is electrically separated from ground and the body of the transformer by at least basic insulation. And, it is separated from the input winding by at least double or reinforced insulation.

For example, toroidal SELV transformers for medical equipment usually have an electrostatic shield tightly wound over the insulation of the primary windings. This shield reduces capacitive coupling between primary and secondary windings, reducing leakage currents at the power-line frequency.

The shield is coated with reinforced insulation to create a reinforced insulation barrier between the primary and secondary windings. The core itself is isolated from the windings by supplementary or reinforced insulation.

Figure 6-V-I *plot of an IS-10 /so-Switch patient* lead-fault *interruptor. Between −7.9 and +7.9 µA (rated ±10 µA), the device acts as a 40-kΩ resistor. Beyond the trip point, the resistance of the /so-Switch increases to approximately 1,000 MΩ at the maximum absolute operating voltage of 325 V.*



Batteries are another convenient alternative for powering medical instruments. This substitution not only ensures inherently low leakage currents, but it makes the equipment highly portable. Take a look at my article, “Battery-Operated Power Supplies” (INK 55).

Whatever the power-supply choice, it’s a good idea to purchase it as an approved OEM assembly. This helps you concentrate on the core of your instrument, rather than dealing with the headaches of designing and constructing supplies which conform to safety standards.

Similarly, designing the instrument to make use of preapproved components helps you considerably in receiving and maintaining safety approval once you embark into the production and sale of your medical product.

You can use components that are not certified by a U.S. Nationally Recognized Testing Laboratory (NRTL) or its equivalent in other countries. However, the assured continuity of safety performance must be investigated for each device. A further complication is that, once you receive approval for your product, any change in any component requires requalification of the complete assembly.

Last, keep in mind that safety standards usually impose special performance characteristics for components such as power cords, switches, line filters, fuse holders, optoisolators, CRTs and displays, and PCBs.

ADDITIONAL PROTECTION

However carefully you design your instrument, absolute safety cannot be guaranteed in the real world. Despite all the FDA-required safety testing and evaluation, medical-device manufacturers still pay high insurance premiums to protect themselves.

It often happens that additional or redundant hardware ensuring safety beyond the minimum requirements is cost effective. It brings concomitant savings in insurance costs due to reduced risk.

Being extra conservative is especially important during prototyping. As an entrepreneur, you probably do not have the legal and financial umbrella of a large corporation to protect you against an unintentional mishap.

My personal preference is to introduce, at the very least, an additional but independent layer of protection against electrical shock at the patient interface.

A practical method accomplishing this is to use Ohmic Instruments’ Iso-Switch patient lead-fault interruptors. These two-lead semiconductor devices can be placed, almost transparently, in series with every patient connection to break the patient circuit when an overcurrent fault develops.

As shown in the V-I plot of Figure 6, an Iso-Switch patient lead-fault interruptor rated at ±10µA acts as a 40-kΩ resistor. Once the Iso-Switch trip point is exceeded, the device presents a negative-slope resistance magnitude equal to the positive slope within the trip boundaries.

The trip time under an overcurrent condition is fast-typically 10 µs. Once the device trips, the resistance of the Iso-Switch increases to approximately 1,000 MΩ at the maximum absolute operating voltage for the device. Once the overload is removed, the device resets automatically.

Various Iso-Switch models are available for different applications, with trip currents ranging from ±1 µA to ±100 mA. Various operating voltage ranges are also offered. You can have ±325 V for 115 VAC instruments, ±720 V for 220-VAC equipment, and for applications where defibrillation protection is desired, models are available which withstand a maximum peak of pulses of ±5 kV for 10 ms.

At prices between $2934 each (one patient connection), the added protection provided by these fault interruptors is most certainly affordable for evaluation prototypes and sometimes even for the final design.

TESTING FOR COMPLIANCE

Although this presentation of medical-device electrical safety standards is by no means intended to replace the actual standards in scope or in content, I introduced many important design requirements covered by the major standards.

With medical equipment, however, designing solely for compliance is not sufficient. The consequences of a malfunctioning device can be so serious that proper performance tests are of utmost importance.

In Part 2, I'll discuss how electrical safety testing should be performed on a prototype medical instrument to make sure that the equipment does not pose a risk of shock to either users or patients, even in the event of foreseeable abnormal conditions. ❑

David Prutchi has a Ph.D. in Biomedical Engineering from Tel-Aviv University. He is an engineering specialist at Intermedics, and his main R&D interest is biomedical signal processing in implantable devices. He may be reached at davidp@mails.imed.com.

REFERENCES

[1] *Intl. Std. CEI-IEC-601-1, Medical Electrical Equip. Part 1: General Req. for Safety,* 1988; Amendment 1, 1991; and 2, 1995.

[2] *Intl. Std. CEI-IEC-601-1-1, Medical Electrical Equip. Part 1: General Req. for Safety, 1. Collateral Std.: Safety Req. for Medical Electrical Sys., 1992.*

[3] *Intl. Std. CEI-IEC-601-2-27, Medical Electrical Equip. Part 2: Particular Req. for the Safety of Electrocardiographic Monitoring Equip., 1994.*

[4] Underwriters Laboratories, *Std. for Safety UL2601-1, Medical Electrical Equip. Part 1: General Req. for Safety, 1994.*

[5] W.S. Woodward, "The Many Uses for Optical Isolators," *Electronic Design,* April 17, 1995, 101-108.

SOURCES

284J **isolation amplifier**
Analog Devices, Inc.
One Technology Way
Norwood, MA 02062
(6 17) 329-4700
Fax: (617) 326-8703

INA110 instrumentation amplifier
IS0107 isolation amplifier IC
Burr-Brown Corp.
6730 S. Tucson Blvd.
Tucson, AZ 85706
(520) 746-1111
Fax: (502) 889-1510

MAX187
Maxim Integrated Products, Inc.
120 San Gabriel Dr.
Sunnyvale, CA 94086
(408) 737-7600
Fax: (408) 737-7194

Iso-Switch fault interruptors
Ohmic Instruments Co.
508 August St.
Easton, **MD 21601**
(410) 820-5111
Fax: (410) 822-9633

IRS

407 Very Useful
408 Moderately Useful
409 Not Useful

FEATURE ARTICLE

Bob Perrin

High-Resolution ADCs

High-precision ADCs save corporations a lot of cost, but are far from a plug-and-play solution for design engineers. Bob goes over many of the gnats that determine which part is best for your situation.

Today, 16-, 18-, 20-, and even 24-bit ADCs are available. These high-resolution ADCs are finickier than their low-resolution cousins. Sometimes, a 24-bit Delta Sigma ($\Delta\Sigma$) ADC is only as effective as a 10-bit converter. Knowing a few basic concepts and common pitfalls helps shorten design times.

In this article, I review crucial ADC concepts, report on four superb converters, and present several precision references. I've geared discussion to practical measurement-system design.

OUR MODEL ADC

Figure 1 shows a generic data-acquisition system. The sensor and low-noise amplifier (LNA) acquire and condition signals ("Microvolt-level Data Acquisition," *INK 68).*

The ADC block converts analog signals into digital data. The precision reference provides a stable standard value for comparing analog signals.

The microprocessor block contains CPU, memory, and glue logic. Typically, this subsystem is the primary source of digital switching noise. The I/O interface contains LCD, keypad, electromechanical drivers and devices. Large current transients originate here.

Commercial ADCs have diverse feature sets. Our model ADC has one channel, measures voltage, and derives its sample rate from an external crystal. The interface signals are:

- Analog Input (Ain)—the signal we desire to convert to a digital word. Ain is considered a single-ended, ground-referenced signal.
- Reference Voltage (V_{ref})—this voltage defines full scale
- Data Out (Dout)-think of this bus as a serial or parallel interface
- Control Bus-this bus contains the signals that start conversions, indicate completed conversions, and direct information flow
- Xin/Xout—the ADC's sample frequency is determined by the crystal

Figure 2 depicts our model ADC.

EFFECTIVE RESOLUTION

There are a number of tradeoffs in designing with high-resolution ADCs. (Resolution is discussed in the sidebar "How Small is Small Enough.")

Effective resolution, also known as noise-free bits, is computed based on the RMS noise inherent to the converter;

Effective resolution = converter resolution $-\log_2$(RMS noise in LSBs)

Consider a megadollar 30-bit ADC. If V_{ref} is 2.5 V, it has a 2.3-nV resolu-

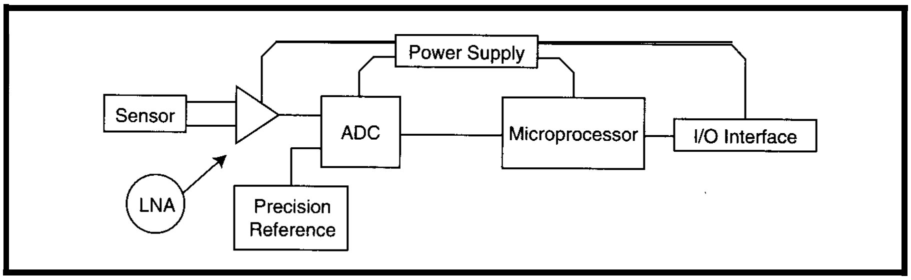


Figure 1—*A data-acquisition system can be divided into independent subsystems. The author focuses on the ADC and precision-reference components.*

tion at Ain. Noise on the ADC dice is likely to be 1-µV RMS – 435 LSB. Since $\log_2(435) = 8.77$, our megadollar 30-bit ADC is reduced to a \$20 21-bit ADC (i.e., 30 – 8.77 ≅ 21-bit ADC).

Effective resolution is a good measure of the ADC's usable bits without a DSP. Additional DSP filtering improves overall system resolution.

A converter's effective resolution depends on sample rate, programmable DSP filters, and PGAs, among other things.

SAMPLE FREQUENCY

Paraphrased, the Nyquist Criterion states: A system only resolves frequency components that are less than half the sample frequency. Interestingly, this concept has been around since 1928 [1].

A system's Nyquist frequency is defined as half the sample frequency. To select a sample frequency, you must determine the maximum signal frequency in the system.

First, find the maximum expected transducer dV/dT, which can then be related to a sinusoid. The maximum dV/dT for a sine is at the zero crossing.

This sinusoidal frequency is the highest frequency component in the system. The sample rate must be at least twice the system's highest frequency.

If a system is measuring something almost static like temperature, select the sample rate by how often you want data posted. A residential thermostat could sample a thermistor at 2 Hz and still be grossly oversampling.

Related to sample rate is aliasing—the phenomenon of a high frequency (above the Nyquist frequency] appearing falsely as a lower frequency (below the Nyquist frequency].

High-resolution ADCs require good antialiasing filters because of their excellent SNR. Even a small signal on Ain can be reflected in the conversion. Any noise above the Nyquist frequency is folded back into the sample band and reflected in the conversion as additional noise.

Consider an ADC with a sample frequency of 20 kHz and a Nyquist frequency of 10 kHz. If an antialiasing filter is placed with its –3-dB point at 8 kHz and the design dictates a –60-dB attenuation at the Nyquist frequency, the filter must drop 57 dB in 2 kHz. That's an unpractical roll off.

Most oversample so a lower-order filter can be used. If we move the Nyquist frequency to 100 kHz (by sampling at 200 kHz), then we only need to drop 57 dB in 92 kHz.

Much LNA intrinsic noise is flicker noise (i.e., 1/f noise) with less power at higher frequencies. Perhaps the filter requirement could be dropped ~12 dB since there is less out-of-band noise to fold back. The system now has an 8-kHz passband and an antialiasing filter that drops 45 dB in 92 kHz.

When choosing a sample rate, keep in mind these tradeoffs:

- the faster the ADC, the more power dissipated
- the faster the ADC, the more it costs
- oversampling can buy additional signal conditioning via DSP
- oversampling can loosen antialiasing filter requirements
- undersampling dooms any design

IMPEDANCE AND CALIBRATION

Input impedance of Ain is also important. Some sample-and-hold circuits swallow small gulps of current during sampling. If you plan to connect a high-Z sensor directly to the ADC (e.g., a thermistor), watch for this loading effect.

High-resolution ADCs have a calibration cycle that can be initiated by the system processor. The calibration runs mandatorily on powerup and periodically later as the system's thermal equilibrium shifts slightly. Low-resolution ADCs don't often have to worry about this calibration.

The time a device runs a full calibration cycle varies from part to part and with resolution and clock speed. Times of 1600 ms are not uncommon.

If the system powers down the ADC, this reset and calibration time can be annoying. The system has to recalibrate the ADC on each powerup, making the system appear sluggish.

Many ADCs have a sleep mode that preserves the last internal calibration information. These modes save power and enable quick restarts.

High-resolution ADCs often have separate analog and digital supply rails. Two independent regulators can be used to guarantee maximum digital switching noise decoupling as long as proper supply sequencing can be assured. A single regulator and an RC decoupling scheme also work.

FOUR COMMERCIAL ADCS

Let's look at four commercial high-resolution ADCs from Analog Devices nd Crystal Semiconductor. There are wo AC converters, the AD7713 and X5506, and two SAR converters, the S5102A and the AD676.

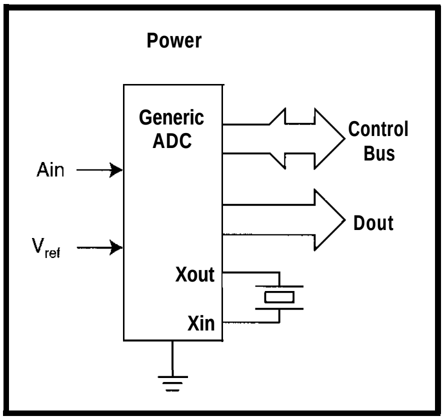


Figure 2—A bare-bones generic ADC is a useful model to keep in mind. Commercial ADCs are just variations on this theme.

Each of these parts has its own niche. Their markets overlap some, but they illustrate the broad variety of high-resolution devices available.

To evaluate these parts, I used the manufacturer's evaluation software and boards to keep noise to a minimum and evaluate the converters-not my prototyping.

However, I analyzed the noise using MathCAD. This way, I could use exactly the same computation methods for the different evaluation systems.

I placed the boards in a box to reduce HVAC air currents and thermal transients. I shut down unnecessary electronic equipment to minimize EMI.

The tests examined three electrical characteristics:

- noise magnitude with Ain = 0
- zero offset
- noise distribution with Ain = 0

I worked with 4096 points per sample set. I collected four sample sets and averaged the statistical results from each set.

Assuming a Gaussian noise distribution, the RMS noise is the standard deviation σ of the sample set.

Noise,., = 6.60

The system offset is the mean of the data set if the inputs were shorted or grounded during data collection. Effective resolution is computed based on RMS noise levels, as previously mentioned. To compute log,, use:

$$\log_2(x) = \frac{\log_{10}(x)}{\log_1(2)}$$

To look at the noise distribution, I averaged four 4096-point FFTs.

If the noise is random, the variance remains the same between sample sets, but the peaks and valleys shift randomly in the frequency domain. The average of the FFTs converges to a fuzzy band if the noise is uncorrelated.

First Notch of Filter and O/P Data Rate	-3 dB Frequency (Hz)	Effective Resolution (bits)							
		Gain of 1	Gain of 2	Gain of 4	Gain of 8	Gain of 16	Gain of 32	Gain of 64	Gain of 128
2									
5	0.52 1.31	22.5 21.5	21.5 21	21	20	20.5 19.5	19.5 18.5	18.5 17.5	17.5 16.5
10	1.57	21	21	2205	20	19.5	18.5	17.5	16.5
12	2.62 3.14	20 20	20	20	19.5 19.5	19 19	18.5	17.5 17	16.5
20			20				18	17	16
50	5.24 13.1	18.5	18.5	15.5 18.5	18.5 15.5	15.5 18	17.5 15.5	15	14.5 16
		15	15						
100	26.2	13	13	13	13	13	12.5	12.5	12.5
200	52.4	10.5	10.5	11	11	11	10.5	10	10

Table 1—*The* AD7713 effective resolution varies *predictably with the PGA gain and first-notch freqency. Reprinted with permission of Analog Devices.*

However, noise from nonrandom sources (e.g., clock feedthrough or bus noise) appears in all the sample sets as a spike on the average of the FFTs.

AD7713 is a 24-bit AC converter with a super feature set. It has two differential channels and one single-ended channel. This device has two current sources (200 µA each) for RTD or bridge excitation and a third smaller current source (1 µA) for thermocouple burnout detection. There is a PGA onboard with gains of 1 to 128.

The AD7713 consumes 3.5 mW of power while sampling and 150 µW in power-down mode. The entire evaluation board consumes 1.3 mA at 5 V. (It powered two 74HC parts, an AD680 2.5-V reference, and the AD7713 with a 2-MHz crystal.) It's a very low-power high-resolution converter.

The PGA and sample rate are programmable via a serial interface. Additionally, the RTD excitation currents, burnout current, bipolar/unipolar operation, 24- or 16-bit word length, and channel selection are set via this serial interface. The same serial interface delivers the ADC codes to the micro.

The AD7713 maintains a 24-bit word that is available to the microprocessor. Internally, the AD7713 probably maintains a longer word for DSP computations.

The effective resolution of this part depends on the sample rate and PGA setting. Table 1 illustrates how effective resolution varies with sample rate and PGA settings.

The evaluation board met or beat the numbers in Table 1. The test results are shown in Table 2. The offset error seemed to drift with time, probably due to the reference (AD680) drifting with temperature.

The AD7713's background calibration mode periodically adjusts the ADC. Using this and a more solid reference (AD780 or LT1019) would probably improve the offset drift.

Figure 3 shows the FFTs of 1.952- and 206-Hz sample sets. The noise in the slower sampled data seems mostly white. The noise in the faster sampled data has a definite shape to it.

The AD7713's feature set, ease of use, and excellent power consumption make it a candidate for many projects. Embedded controller applications, data loggers, or hand-held instrumentation are excellent applications for this part.

Sample rate	PGA	Noise$_{RMS}$	Noise$_{p\text{-}p}$	Offset error	Effective resolution (bits)
AD771 3					
1.952 Hz	1	637 nV 4.953 LSB	4.2 µV 28.2 LSB	-436 nV -2.9 LSB	21.9
1.952 Hz	128	96 nV 82.953 LSB	631 nV 542.6 LSB	-1.2 µV -1075 LSB	17.6
206 Hz	1	1698 µV 11400 LSB	11200 µV 75260 LSB	-1887 µV -12670 LSB	10.5
206 Hz	128	12.2 µV 10520 LSE3	80.8 µV 69400 LSB	-1.2 µV -176 LSB	10.6
CS5506					
20.2 Hz	N/A	16.9 µV 3.5 LSBs	112 µV 23.4 LSBs	8.7 µV 1.8 LSBs	18.2
CS5102A					
20 kHz	N/A	38 µV 0.5 LSBs	250 µV 3.2 LSBs	-181 µV -2.4 LSBs	15.5
AD676					
96 kHz	N/A	85 µV 0.6 LSBs	562 µV 3.7 LSBs	283 µV 1.9 LSBs	15.4

Table 2—*These empirical results offer a baseline to compare other converters and systems against.*

Several sister parts in the AD77xx family have varying feature sets. If high resolution and slow sample rate are what you need, consider them.

The CS5506 is a 20-bit AI: converter. This part has four pseudodifferential channels, flexible supply voltages, +V$_{ref}$ and –V$_{ref}$, internal or external references, and low power dissipation (1.7-3.2 mW). A simple serial bus provides the data.

The CS5506 is designed for use with a 32.768-kHz watch crystal, yielding a 20.2-Hz sample frequency. Other clock frequencies may yield a 100-Hz sample frequency, but optimal DSP-filter characteristics are maintained with the recommended watch crystal.

The four pseudodifferential channels use a common –Ain and have four +Ain pins. –Ain is not referenced to ground. This system works well.

Unlike the AD7713, all of the CS-5506's features are controlled with hardware pins. There's no control register to program.

The CS5506 offers true unipolar/bipolar operation. In bipolar mode, it mirrors the difference of +V$_{ref}$ and –V$_{ref}$.

How Small is Small Enough?

The simplest method to determine ADC resolution is to divide the desired input range-say, 600 µV—by the desired system resolution-say, 40 nV, which yields the 15,000 codes required. You then find the nearest higher integer power of two: 2^{14} = 15384, n = 14 (see Table i) to get the minimum resolution converter needed.

You can also start with the desired percentage change in sensor fluctuation. If the system measures temperature ranging 0-100°C and must resolve a 1/1000°C change, the desired resolution as a percentage is:

$$\frac{0.001°\text{C}}{100°\text{C}} \text{ x } 100\% = 0.001\% \text{ resolution}$$

A 17-bit ADC resolves the temperature change, but the nearest common commercial ADC is 18 bits (see Table i).

SNR is another resolution characteristic. Any converter has an inherent ±½ LSB uncertainty-the quantization error-associated with a conversion. The ideal SNR of an ADC is defined as:

$$\frac{\text{full-scale RMS sine}}{\text{RMS quantizationerror}}$$

Commercially available	2^n	Codes	Res (ppm)	Res (%)	Ideal S/N (dB)
					74
yes	2^{12} 2^{13}	4096 8192	[illegible]	0.024 0.012	80
	2’4	15384	31	0.0065	86
	2’5	32768		0.0031	92
					98
yes	2^{16} 2^{17}	65536 131072	15 7.6	0.0015 0.00076	104
yes	2^{18}	262144	3.8	0.00038	110
	2'9	524288	1.9	0.00019	116
yes	2^{20}	1048576	0.95	0.000095	122
	2^{21}	2097152	0.48	0.000048	128
yes	2^{22}	4194304	0.24	0.000024	134
	2^{23}	8388608	0.12	0.000012	140
yes	2^{24}	16777216	0.06	0.000006	146

Table i--Depending on your application, you *can express resolution in several* convenient units.

From this, you get an SNR = 6.02n dB + 1.76 dB[2].

An ADC can't do better than its ideal SNR, but it can do worse. Spending thousands on a liquid-helium-cooled LNA and transducer to get a 200-dB SNR is a waste if a 16-bit ADC (98-dB best case SNR) captures the data.

A converter's effective SNR improves with oversampling and filtering, which is how a AC converter overcomes its single-bit converter's ridiculous 7.8-dB SNR (6.02 x 1 dB + 1.76 dB).

Sample sets were taken using bipolar mode, and their results are in Table 2.

The reference on the Crystal evaluation board is a 2.5-V LT1019. The offset remained reasonably constant over the data sets.

The CS5506 has an internal 2.5-V reference. The data sheet indicates a suitable external reference might provide increased performance, and the evaluation board ships with an external reference.

The FFTs of the samples came out flat with white system noise. The CS5506 spectrum came out so nice, I ran the test several times. Each time, I got similar excellent results.

The CS5506 is a neat little part. It has low noise, low cost, low power consumption, four channels, and high resolution. Its family has several parts with varying features and resolutions.

The CS5102A is a 20-kHz, 16-bit SAR converter. The two single-ended inputs are referenced to analog ground. The power consumption is 44 mW. This part sports a true unipolar/bipolar input scheme.

Its features are controlled through hardware control pins. It communicates via a serial bus.

The only quirk to watch for is power-supply sequencing. If two separate positive supplies are used, one is for digital and one is for analog. The analog supply must come up before the digital. If only one supply is used, an RC filter can decouple the digital switching noise.

For the experiment, V_{ref} was supplied with a 2.5-V LT1019. The part was run in bipolar mode (± 2.5 V). The results are in Table 2.

The FFT revealed that the noise was white. The spectrum was very flat and no noise shaping was apparent.

The C55102A is simply an SAR converter with the sample rate being determined by crystal frequency.

The CS5 102A is an old friend of mine. It's a good and reliable part. The CS5101A is a similar but faster (100-kHz) part. Both offer a fast, simple, 16-bit SAR solution.

The AD676 is a 100-kHz, 16-bit SAR single-channel ADC with a parallel data bus. Unfortunately, the data bus does not tristate. You must provide an external tristate latch if you connect it to an controller's data bus.

The AD676 likes ±12 and +5 V. Its top power dissipation is 480 mW.

The evaluation board came with an AD586 5.0-V high-precision reference. This reference is very stable with a 2-ppm/°C temperature coefficient. Offset drift wasn't a problem. The evaluation board normally ships with a 10-V reference. The FFT revealed the noise was white.

The AD676 is a 28-pin package. A sister part, the AD677, has a serial output. The AD677 is only a 16-pin DIP. If you have spare power and need a small package, it's worth exploring.

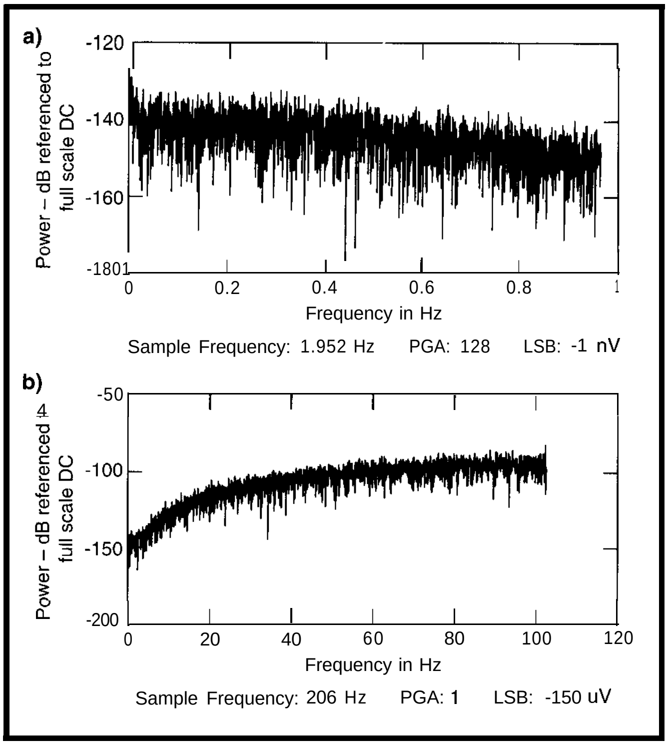


Figure 3-h *contrasting the FFTs, clearly the besf noise performance is obtained at slow sample rates. a) At slow Fs, the noise appears to be preffy white with a slight downward trend at higher frequencies. b) At high Fs, the noise has a definite shape. This is due to the noise-shaping DSP techniques employed by the ΔΣ converter.*

PRECISION REFERENCE

The world's best ADC is only as good as its voltage reference. Selecting a low-noise, low-drift precision reference is as important as selecting the proper ADC.

Figure 4 offers four reference circuits. Figure 4a shows a simple 2.5-v reference with some noise filtering. A band-gap reference with a negative reference voltage is in Figure 4b. Figure 4c illustrates how a nonstandard reference voltage may be obtained.

Occasionally, it's good to have an ADC referenced asymmetrically. Then, Figure 4a could provide the $+V_{ref}$ and Figure 4c the $-V_{ref}$.

The system could measure small voltages above and below zero (e.g., ADC, LNA, and sensor offsets). In addition, it could measure large positive voltages (e.g., transducer signals).

However, two references means two noise sources. Since they are uncorrelated, they add in an RMS sense. You must weigh the extra system noise against the features gained.

Noise on the ADC's V_{ref} pin is reflected in the conversion exactly like noise on the Ain pin. This concept is important but often neglected. Choosing a noisy reference can spell doom for a high-resolution ADC.

Zener diodes, which suffer from avalanche noise and large temperature coefficients, are not a good choice for precision referencing.

Adding an op-amp as a buffer (Figures lc and ld) increases noise. In Figure 4c, the op-amp must maintain a low-impedance reference voltage to the ADC, but in Figure 4d, the op-amp is not needed as a buffer and is an undesirable source of error and noise. As a rule, keep reference circuits simple to minimize component count and noise sources.

Figure 4d is useful if the ADC takes its reference at the LT1019's output and a thermistor or other transducer requiring a controlled voltage is tagged onto the output of the unity-gain buffer. This keeps noise from long sensor wires buffered from the reference pin.

A characteristic related to noise is output drift with temperature change.

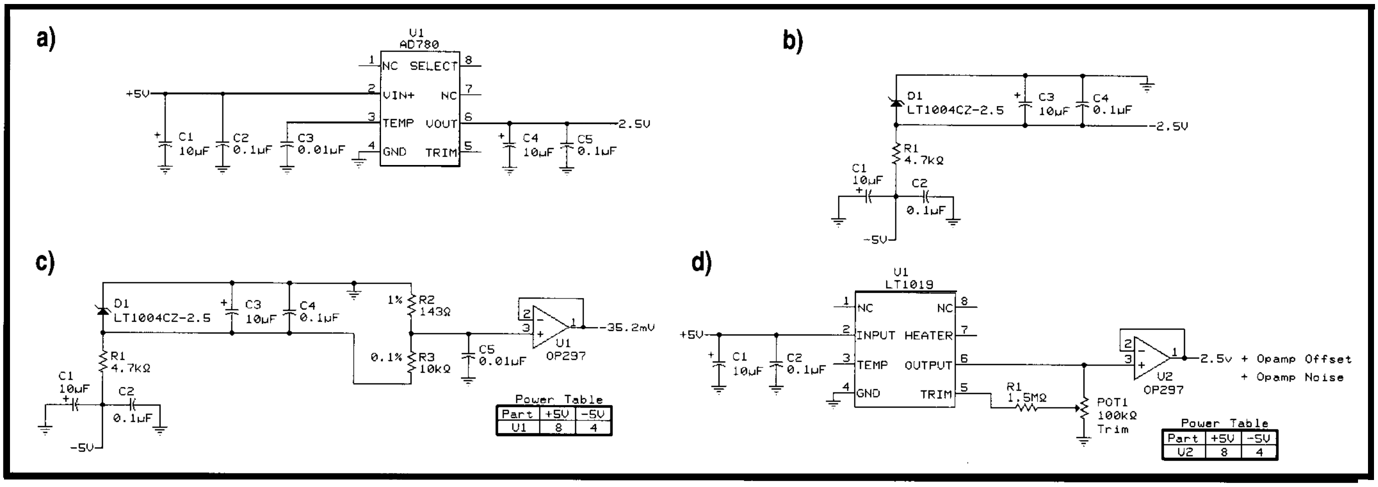


Figure 4—*The precision-reference* **circuit is as pivotal** *to* **system precision as the** *ADC itself. a) The* **AD780 provides a** *stable 2.5* **V** *(or 3.0 V) reference voltage. b) The LT1004 bandgap can be used as a zener* **diode.** *This type of reference is useful for deriving negative voltage. c)* **A** *precision voltage divider and low offset follower* **with** *a precision reference generates* **stable** *nonstandard reference values. d) Superfluous buffers increase noise and absolute error and should be avoided.*

This characteristic is usually specified in ppm/°C (see Table i in the sidebar, "How Small is Small Enough").

A 16-bit converter has a resolution of 15 ppm. A precision reference with a 20-ppm/°C drift eats ADC codes with only slight thermal shifts on the PCB. Keep the LNA, ADC, and reference shielded from thermal shifts.

Older band-gap references are not well-suited to high-precision systems. The LT1004-2.5 is an older bandgap with a 2.5-V reference useful in 8- and 12-bit systems. However, its accuracy (f20 mV), stability (20 ppm), and intrinsic noise (60-120 μV_{p-p}) make it borderline for high-resolution systems. The LT1019, however, is a third-generation band-gap reference with excellent characteristics [3].

During my research, I came across an interesting part-the LTZlOOO. This device seemingly maintains the phenomenal temperature stability of 0.05 ppm/°C. It has an on-chip heater (resistor) and temperature sensor (base-emitter junction).

The LTZlOOO data sheet indicates that the device must be shielded from thermal gradients on the PCB. The device pins must be kept isothermal and shielded from 'air currents. These are excellent tips for *any* precision-reference design.

Careful selection of resistor material is essential. As the LTZlOOO data sheet states, "Wire-wound resistors usually have the lowest thermocouple voltage, while tin-oxide-type resistors have very high thermocouple voltage. Film resistors, especially Vishay precision film resistors, can have low thermocouple voltage" [4].

THE FINAL TUNING

The advent of discrete, inexpensive high-resolution ADCs has been a grand windfall for instrumentation engineers. They enable engineers to acquire data with a degree of precision not feasible until recently.

However, all the classic digital-sampling issues-noise aliasing, anti-aliasing filters, and system noise—must be addressed when employing these high-precision ADCs.

Modern components lower the cost-performance ratio of system designs, but they're not magical plug-and-play solutions. With higher demands on electronic equipment, design engineers have as many details to tend to as ever. And, the devil is in the details. ❑

I wish to thank Crystal Semiconductor and Analog Devices for their help.

Bob Perrin designs nanovolt mixed-signal systems for Decagon Devices. He also designs the electronics for Decagon's agricultural products. You may reach Bob at nanobob@turbonet.corn or nanobob@decagon.com.

REFERENCES

[1] Oppenheim and Schafer, "Frequency-Domain Representation of Sampling," *Discrete-Time Signal Processing,* Prentice Hall, Englewood Cliffs, NJ, 82-87, 1989.

[2] Analog Devices, "High-Resolution Data Conversion," *Analog-Digital Conversion Handbook,* Prentice Hall, Englewood Cliffs, NJ, 542, 1986.

[3] Linear Technology, *Linear Technology Databook, 1990.*

[4] Linear Technology, "LTZ1000/LTZ1000A Ultra Precision Reference," *Linear Technology Databook, 3-9-3-15, 1990.*

SOURCES

AD7713, AD676, AD680, AD677
Analog Devices
P.O. Box 9106
Norwood, MA 02062-9 106
(617) 329-4700
Fax: (617) 461-4261

CS5506, CS5102A, CS5101A
Crystal Semiconductor Corp.
4210 S. Industrial Dr.
Austin, TX 78744
(512) 4457222
Fax: (512) 445-7581

LT1004, **LT1019,** LTZlOOO
Linear Technology
1630 McCarthy Blvd.
Milpitas, CA 95035-7487
(408) 432-1900
Fax: (408) 434-0507

I R S

410 Very Useful
411 Moderately Useful
412 Not Useful

In-Circuit Emulators

Part 3: Low-Cost ICE

FEATURE ARTICLE

Graham Moss, Ross McMillan & Ken Mardle

Wrapping up this series of articles, the authors look in on a specific low-cost ICE—Philips' PDS51 emulator. Hardware, software, and specific debugging techniques offer ways to get around tricky debug tracing.

In Part 2, we discussed sophisticated debugging tools including monitor-based debuggers and true in-circuit emulators (ICEs).

In this final article, we provide an in-depth look at the design and implementation of PDS51, a low-cost ICE development system from Philips.

WHAT IS PDS51?

The PDS51 emulator system is a full-featured ICE for Philips' 8xC51 microcontroller family. It provides a compact (110 x 130 x 42 mm) board-level product supporting a wide range of device variants for about $1000.

The emulator connects to the target via a short, flat ribbon cable terminating in a device footprint connector. It connects to a host via a serial RS-232 cable carrying communications at up to 115 kbps. Power comes from a standard calculator-style plug-pack DC power supply.

The emulator has two stacked PCBs as shown in Photo 1. A motherboard accommodates common ICE elements (i.e., power regulation, host interface, control microcontroller, memory, and logic). A daughterboard houses the execution microcontroller and target interface and customizes the system for a specific group of 8xC51 derivatives. A range of footprint connectors supports DIP, PLCC, and other target-package types with varying numbers of pins.

PDS51 includes a built-in command-line-driven interface (compatible with the older Philips SDS emulator). Any host computer capable of running an ASCII terminal emulator can be used, but PDS51 is typically used with integrated development environment (IDE) software for PCs.

Although PDS51 was designed to support development using Philips devices, it isn't limited to this role.

PDS51 HARDWARE

Since PDS51 is a true ICE, it offers maximum assistance without lost resources, limited flexibility, or non-real-time operation. And, because it's an ICE, it needs a bondout device for emulation. It also operates the target micro in single-chip, externally expanded, and mixed-memory modes.

Figure 1 shows the PDS51's basic architecture. PDS51 provides 64 KB of emulation memory for program code, the maximum addressable by an 8xC51 device without bank-switching. External accesses to data memory are routed to target-system memory devices for proper operation of memory-mapped peripherals and bank-switched data memory.

Not only does PDS51 have unlimited standard code-address breakpoints, it conditionally breaks on external stimuli from test equipment or signals

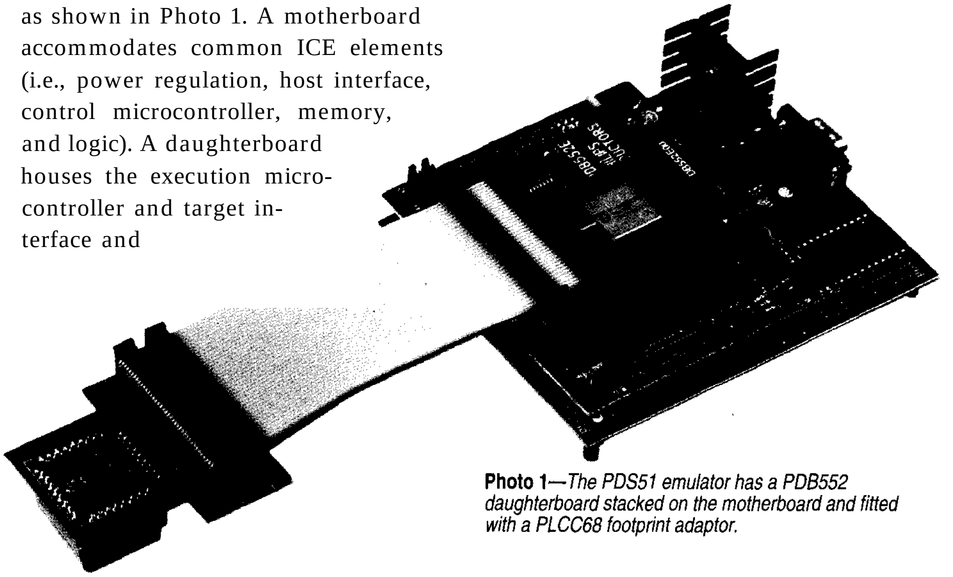


Photo 1—*The PDS51 emulator has a PDB552 daughterboard stacked on the motherboard and fitted with a PLCC68 footprint adaptor.*

from the target system. It also triggers external test equipment on the basis of fetches from any code address.

PDS51 traces execution in real time on a cycle-by-cycle basis. External-signal states from the target systems in the trace record give a properly synchronized logic-analysis capability.

To achieve this functionality, the 64 KB of emulation code memory has an another 64 KB of memory devoted to breakpoints, triggering, and trace control. There are 8 independent control bits for each code-fetch address.

There's no limit on the number of breakpoints, trigger points, and trace qualifiers which can be set. A further 128 KB of memory is shared between a trace buffer and a small additional buffer for emulation kernel code.

The trace buffer is 28K words deep, and each word is 32 bits wide. It updates with a new trace record every cycle. Each record includes 16 code-fetch address bits, 4 status bits, and 12 channels of state information given by the user to a daughterboard connector.

Although the trace buffer is relatively shallow, trace recording is qualified on an instruction-by-instruction basis and by external control signals. Since the trace buffer records code-fetch addresses and not code, tracing self-modifying code isn't supported.

Supporting logic used in Philips high-speed CMOS PLDs gives complex buffering, multiplexing, and routing logic. These features make the two fast byte-wide memory devices resemble four slower but independent memories of widely differing organization.

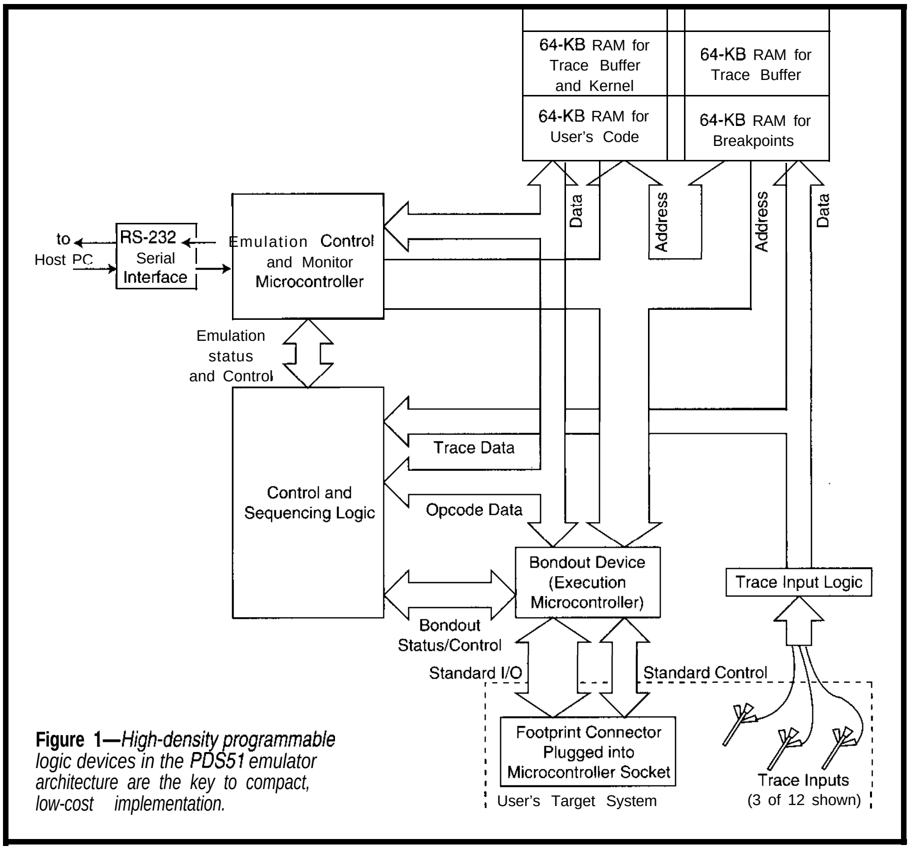


Figure 1—*High-density programmable logic devices in the PDS51 emulator architecture are the key to compact, low-cost implementation.*

Supporting logic also provides port regeneration, control of breakpoint and tracing subsystems, and timing. These features enable the control micro to tightly manage the execution micro.

A control program in the 8xC528 interacts with the bondout using code fragments stored in emulation-kernel memory. This interaction gives the 8xC528 access to the contents of the SFRs, internal data memory in the bondout device, and external data memory in the target system without consuming bondout resources.

PDS51's real-time emulation aspects are as transparent as possible. Execution when starting, stopping (at a command or breakpoint], and single-stepping suffers no lost cycles, insertion of extra cycles, unusual status conditions, nor unexpected states on target port pins. There also aren't timer/counter delays during stop-start emulation.

PDS51 achieves such transparency in all but one respect. You can't step through a serial output routine which writes to the SBUF register and waits

Using Target to Extend Emulator Power

When debugging, it's important to be able to break execution when certain values appear in certain registers or memory locations. For example, when a variable is being trashed for no apparent reason, you might want to instruct your emulator to "run until location 30h = 80h."

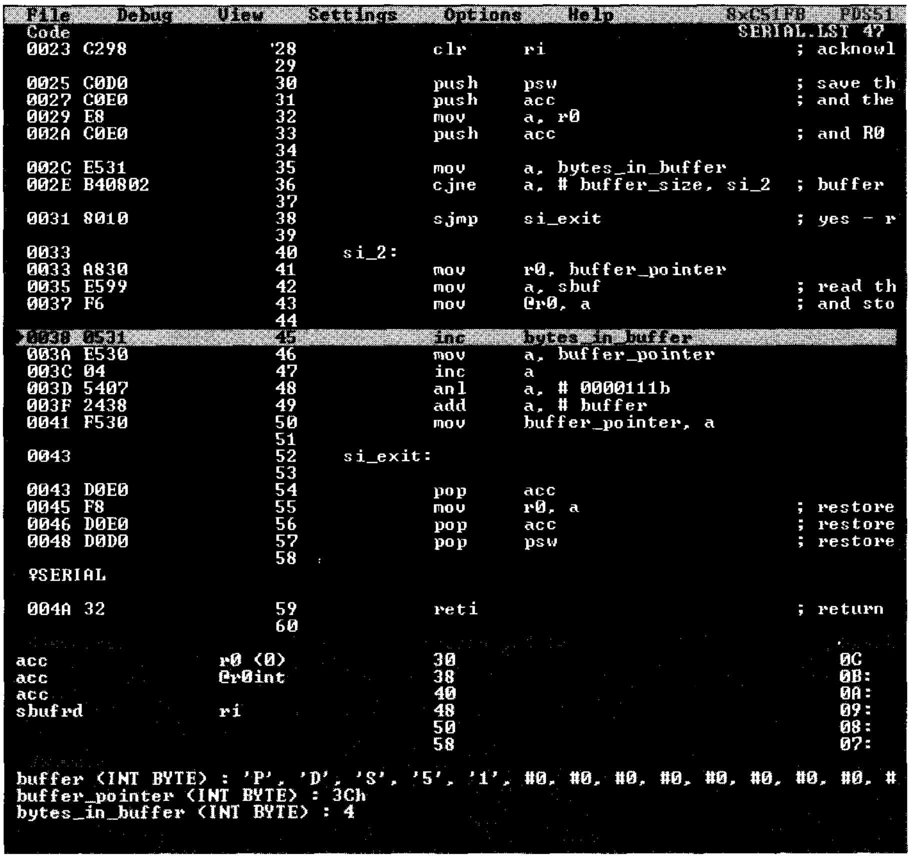
Some emulatois let you specify an address where execution stops and a condition is checked. If the condition isn't met, execution restarts. Some let you pass through an address a number of times before checking.

This powerful technique slows execution because the emulator halts execution and reads and compares the target-variable contents each time the address is struck.

However, it's possible to check for a trashed variable at near-full emulation speed. All programs normally have a portion of code executed regularly and frequently (e.g., a timer interrupt routine). Just add this code fragment:

```
        mov     a, variable
        cjne    a, # 80h, temp_1
        nop
temp_1:
```

If you place a breakpoint on nop and run the program, execution breaks the next time this code fragment executes after the variable is corrupted. By inspecting the prior instructions in the emulator's trace buffer, you can see where the location was overwritten.



```
 File   Debug   View   Settings   Options   Help          8xC51FB  PDS51
 Code                                                      SERIAL.LST 47
 0023 C298                   clr     ri                    ; acknowl

 0025 C0D0                   push    psw                   ; save th
 0027 C0E0                   push    acc                   ; and the
 0029 E8                     mov     a, r0
 002A C0E0                   push    acc                   ; and R0

 002C E531                   mov     a, bytes_in_buffer
 002E B40802                 cjne    a, # buffer_size, si_2 ; buffer

 0031 8010                   sjmp    si_exit               ; yes - r

 0033              si_2:
 0033 A830                   mov     r0, buffer_pointer
 0035 E599                   mov     a, sbuf               ; read th
 0037 F6                     mov     @r0, a                ; and sto

>0038 0531                   inc     bytes_in_buffer
 003A E530                   mov     a, buffer_pointer
 003C 04                     inc     a
 003D 5407                   anl     a, # 0000111b
 003F 2438                   add     a, # buffer
 0041 F530                   mov     buffer_pointer, a

 0043              si_exit:

 0043 D0E0                   pop     acc
 0045 F8                     mov     r0, a                 ; restore
 0046 D0E0                   pop     acc                   ; restore
 0048 D0D0                   pop     psw                   ; restore

 ?SERIAL

 004A 32                     reti                          ; return


acc              r0 (0)          30                            0C
acc              @r0int          38                            0B:
acc                              40                            0A:
sbufrd           ri              48                            09:
                                 50                            08:
                                 58                            07:

buffer (INT BYTE) : 'P', 'D', 'S', '5', '1', #0, #0, #0, #0, #0, #0, #0, #0, #
buffer_pointer (INT BYTE) : 3Ch
bytes_in_buffer (INT BYTE) : 4
```

Photo **2**—*Here's* a typical *debug session with the PDS51 IDE. A serial communications driver is being debugged at the assembly language source-code level. But, the IDE is equally at home with high-level languages.*

for the TI status bit to go high, indicating data was sent. With a bondout, the TI bit never goes high if it stops while the SBUF contents are being serialized.

Extensive use is made of multiplexing and other time-consuming logic techniques. PDS51 is guaranteed to run at 12 MHz, but operates at up to 16 MHz at room temperature.

Although PDS51 has a built-in command-line-driven interface, its full power is only realized in association with an IDE Philips developed for PCs.

TILED DISPLAY

Photo 2 shows a typical PDS51 display screen. It shows the code being debugged, micro resources, internal memory data, the stack, and user data in memory areas. The IDE optionally displays the external data area if the micro operates in an externally expanded mode or has auxiliary memory.

The IDE provides two basic screen layouts. The code pane in Photo 2 is best suited to source-level debugging. A narrower code pane occupying the full screen height is useful for debugging at the machine-code level.

The code pane displays source code or disassembled assembly code (a symbolic disassembly if user symbol data is available). You can switch between these two modes regardless of layout.

PDS51's two data-display panes are fully user-configurable in terms of the data displayed and its format.

FLEXIBLE FORMATTING

Many debuggers fall short when it comes to accessing many of the target's data resources. With internal and external data memory, tools tend to be limited in how they format and display data-often supporting hex formats and little else. For SFRs, major problems arise where the SFRs are not simple single-byte read-write registers.

The PDS51 IDE supports display or entry data formatting and access to underlying data structures. Scalar data can be in binary, octal, hex, ASCII, unsigned decimal, or signed (2's complement) decimal. Scalar data can be 1-4 bytes in length, and multibyte values can be Little- or Big-Endian.

The IDE also supports display and entry of four-byte IEEE-format floating-point numbers. Structured data is displayed as arrays of scalars. Strings are both null-terminated or length-byte prefixed forms.

The PDS51 IDE provides a powerful mechanism for accessing data and supports the 8xC51 native addressing

Debugging Impure Code

We generally think of microcontroller programs as collections of instruction codes. But, we [or our compiler) sometimes embed data in the code using look-up tables, constants, and strings.

In Part 2, we discussed a problem with monitor debuggers replacing code with a `jump` to implement a breakpoint. Another problem rises if the assembly-language programmer or compiler follows a call to a routine with literal data to be used by that routine:

```
call   display-message
db     'Error', 0
call   display-message
db     CR, LF, 0
```

The message-display routine reads the string from the code segment, using the return address on the stack to locate it. It then adjusts the return address on the stack so control returns to the instruction following the string.

Any monitor-based debugger or emulator automatically stepping over the `call` mistakenly places a breakpoint on the string data. So, the breakpoint is never struck and execution runs away at full speed. But, in the case of a monitor, the `jump` it uses for a breakpoint also corrupts the string data.

A similar situation can occur if PDS51 is configured to insert `nop` instructions into the code at breakpoints.

Moral: if you use this technique, don't try to step through that part of the code.

Listing 1—*A fragment of C source code illustrates the challenge presented by the requirement for single-stepping of high-level languages to execute at real-time speeds without interruption.*

```
100 if (function_a==1) or (function_b==2) or (function_c==3) {
101     index ++ ;
102 }
103 else {
104     index -- ;
105 }
```

modes. You can display and modify register R0 and the memory location in internal or external memory pointed to by R0 using `@R0int` and `@R0ext`.

The IDE handles multibyte registers such as DPTR and the timer/counter reload registers. For timer T1, the IDE defines not only the standard 8-bit SFRs (i.e., TH1 and TL1) for display and modification, but also a 16-bit concatenation which can be displayed and modified like a single register.

The IDE also knows about data resources with complex or unusual storage allocations. The 10-bit ADC of the 8xC552 device has the eight MSBs in one 8-bit register and the two LSBs at the MSB end of another. The PDS51 IDE defines a resource called ADC and displays it like a single 10-bit register.

With doubled-up registers, one register is accessed by a read operation and the other by a write. The IDE treats and displays the registers' contents as logically separate entities.

TRACE SUPPORT

A pop-up window accessing PDS-51's trace buffer displays the address of each trace entry, a disassembly of the machine code at that address, the machine-cycle type being executed, and 12 user-selected logic states.

A cursor scrolls through the trace buffer, updating a display of delta time relative to a reference point in the buffer. The IDE enters a crystal frequency for this time calculation.

You can scale the apparent execution speed for any crystal frequency. Although you can't run the emulator at the 40-MHz speed of an 8xC750, you can view the trace as if you were.

A second pop-up window displays the trace-buffer contents in a statistical execution profile. You can set each bin size and address range covered to identify execution bottlenecks and other performance-related problems.

EXECUTION CONTROL

Control over program execution is important in any debugger. Philips offers consistency in both assembly- and high-level language debugging, without losing the PDS51's transparent real-time execution capabilities.

Transparent real-time operation was also important for the IDE, especially for debugging at the source level. Single-step operations in a high-level language program execute at real-time speed until the target step address is reached, regardless of the underlying machine code's complexity.

Listing 1 shows a C program fragment. If you're at line 100 and Step Over is pressed, you'll arrive at line 101 or 104, depending on the tests' outcome. The function calls in the Boolean test may involve critical real-time operations and should be executed at full speed with background interrupt processing.

When the IDE encounters this situation, it scans the generated machine code associated with the current high-level language statement for all possible departure points (i.e., places where control could transfer outside the code generated for this statement).

If any departure point is a jump outside the region, it's covered with a hardware breakpoint at the destination address. If it's a call to another routine, no breakpoint is set since control eventually returns to this region. This approach critically depends on PDS-51's 64K available breakpoints, and it supports high-level language statements of almost any complexity.

The IDE uses different breakpoint channels in the PDS51 emulator, so IDE-placed breakpoints (automatically removed after use) won't tangle with user-placed ones (which can be disabled and reenabled at will and left in place for long periods of debugging).

Some high-end emulators provide additional hardware to implement functions such as breaking on a particular value. See sidebar

BREAK BEFORE AND AFTER

If you've used software-based debuggers and monitor-based systems, you've seen situations where execution stops at a breakpoint before an instruction at that address executes (i.e., "break-before" operation).

Hardware-based emulators, on the other hand, use address-recognition logic which compares the code-fetch address with target breakpoint addresses. So, the breakpoint isn't struck until the fetch occurs. After the in-

execution stops and the PC points to the instruction following the breakpoint (i.e., "break-after" operation).

Since a break-after operation can be confusing, users typically prefer break-before operations. So, PDS5 1 optionally implements the break-before action while operating in a break-after fashion.

PDS51

instruction at any breakpoint with a instruction. The emulator executes nop when the breakpoint is hit, decrements the program counter, and replaces nop with the original instruction.

Since one additional instruction cycle executes for every breakpoint struck, affecting the real-time behavior of counters and other peripherals, this feature can be switched off. If it is enabled, any inserted nop is tagged in the trace buffer and displayed as "inserted nop" in a trace listing.

See "Debugging Impure Code" sidebar for further problems.

ADDITIONAL TOOLS

You can debug code with the PDS51 IDE no matter how it was generated. Philips supports source- and machine-level code debugging, as well as a wide range of commercial assemblers and compilers.

This IDE technical-design challenge was aided by a standard for 8xC51 debug object files called OM F5 1, which was created by Intel. Most major 8xC5 1 program development vendors support the **OM F5 1** format.

To provide adequate support for source-level debugging of assembly-language code, the PDS51 IDE uses the listing file produced by several popular assemblers. In addition to generic Intel hex, binary, and OM F **5 1** file formats, the PDS5 1 IDE supports several code-generation products, including all major 8xC51-targeted compilers.

PDS51 is a powerful debugging tool in the form of a fully-featured ICE. A multiplexed bondout, high-density memory, and programmable logic components implementing the emulation hardware combine with innovative IDE software to offer an emulation system with substantial debugging capability at an affordable price. ❑

Graham Moss is a design engineer with the applications laboratory of Philips New Zealand, which designs and markets a variety of low-cost development tools for microcontrollers, He may be reached at graham8 pds. co.nz.

Ross McMillan and Ken Mardle are design engineers with Applied Digital Research, a New Zealand company specializing in embedded systems solutions and development tools. You may reach Ross and Ken at info@adr. co.nz.

SOURCE

PDS51 and IDE software
Philips Semiconductors
811 E. Arques Ave.
Sunnyvale, CA 94088-3409
(408) 991-5207
Fax: (408) 991-3773
http://www.he.net/~pds/

I R S

413 Very Useful
414 Moderately Useful
415 Not Useful

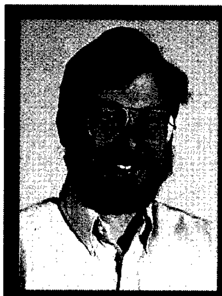
DEPARTMENTS

Tuning Up

FIRMWARE FURNACE

Ed Nisley

Part 2: Zerobeat Firmware



With integration it gets harder to figure out what a circuit is doing. This month, Ed takes a look inside Zerobeat's micro. He wants to know how the firmware is measuring audio frequency and converting it into a moving-dot LED display.

back when Morse code represented state-of-the-art digital communications, you could figure out what a circuit did by looking at it. Each component was visible to the naked eye and sprouted only a few connections.

Contrast that situation with your latest electronic gadget. Assuming you can get the case open, you find one or two flat-black epoxy blocks sprouting dozens—if not hundreds—of leads. Tracing the circuitry reveals only the power supply and, perhaps, the display and switches. Nearly all the good stuff takes place within that epoxy—safe from prying eyes.

Even the Zerobeat hardware I introduced last month presents a challenge because the signals vanish into a single-chip microcontroller. Datasheets don't help at all because what an Atmel 89C1051 (or any other computer) does with the signals depends on the firmware inside. Unless you have the program, you can't tell what happens to the players!

This month, I describe how the firmware measures audio frequency and converts it into a moving-dot LED display. If the thought of assembly-language programming fills you with dread, take heart. You'll see a trick that saves lots of sweat.

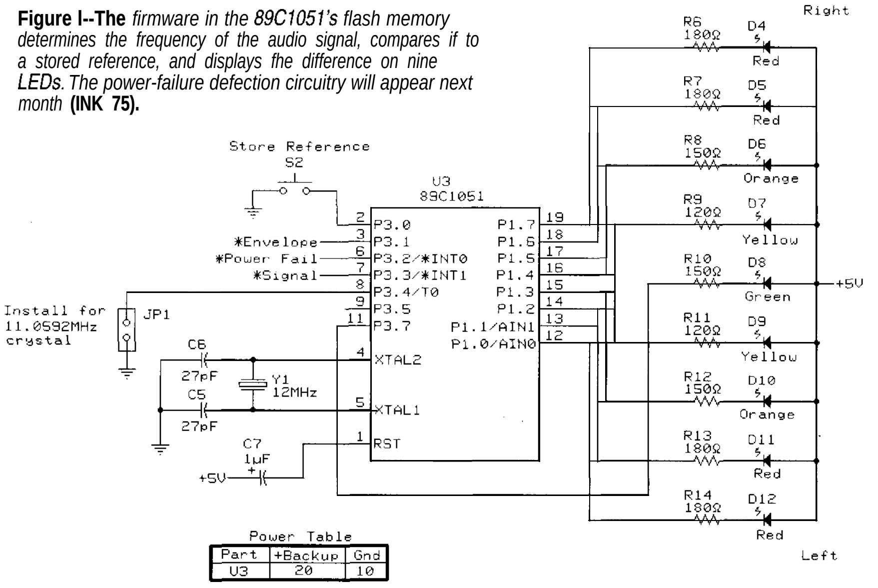
MEASURING CYCLES

Morse code consists of long and short pulses separated by silent intervals. Fixed timing relationships between these elements make the code self-clocking, allowing transmission and reception at any speed. The Amateur Extra license requires proficiency at 20 wpm-well beyond my (current!) ability.

Experienced operators can hold Morse conversations at speeds near the 35 wpm. I've overheard and clocked them with an oscilloscope. The shortest pulses from those expert fingers last about 30 ms and sound like the chittering of small mammals. Although I can't find the exact reference right now, I believe the all-time speed record lies near 70 wpm-a 17-ms dit!

Even at those speeds, each dit includes a dozen or more 700-Hz cycles. That should be enough time for the firmware to synchronize with the signal and measure its frequency.

However, a little arithmetic is in order before we get into the coding. We can determine frequency directly-by counting the number of cycles in a given length of time-or indirectly—by measuring the duration of a single cycle. Our chosen method depends on the required accuracy, available time, and affordable hardware.



Part	+Backup	Gnd
U3	20	10

Figure l--The *firmware in the 89C1051's flash memory determines the frequency of the audio signal, compares if to a stored reference, and displays fhe difference on nine LEDs. The power-failure defection circuitry will appear next month* **(INK 75).**

Direct frequency measurement works well with continuous signals that allow a protracted measurement time. Without excruciatingly clever techniques, achieving 1% accuracy (10 Hz at 1 kHz) requires about 100 cycles. In our situation, we must make the measurement over a burst of only a few dozen cycles.

Listing 1--This *routine runs Timer 0 between two falling edges of the audio input signal. The accumulated count gives the audio period in microseconds if the CPU uses a 12-MHz count. If no transitions occur, the Timer 0 interrupt handler in Listing 2 c/ears TR0 and forces a zero return value.*

```
GetPeriod
* set up Timer 0
          CLR  TRO                     * make sure counter is off
          CLR  TFO                     * shut off interrupt flag
          MOV  THO,#=MAX_PERIOD        * set max possible count
          MOV  TLO,#MAX_PERIOD
          SETB ETO                     * enable timer interrupt
* start timer, wait for falling edge
* if timeout. Timer 0 interrupt clears TRO and
* we exit with a zero
          CLR  INT_AUDIO               * clear edge detector
          SETB TRO
?gpstart  JNB  INT_AUDIO,?gpstart      * wait for input edge
          JBC  TRO,?gp1                * was timer still running?
          SJMP ?gptimeout              * no, timeout
?gp1
* reload timer and wait for next falling edge
          MOV  THO,#=MAX_PERIOD        * set max possible count
          MOV  TLO,#MAX_PERIOD
          SETB TRO
          SETB ETO                     * enable timer interrupt
          CLR  INT_AUDIO               * clear edge detector
?gpstop   JNB  INT_AUDIO,?gpstop       * wait for input edge
* stop and read Timer 0
          JBC  TRO,?gprun              * was timer still running?
?gptimeout
          CLR  A                       * was off, force zero return
          MOV  B,A
          SJMP ?gpdone
?gprun
          MOV  A,TLO                   * count - max count = elapsed
          CLR  C
          SUBB A,#(MAX_PERIOD+FINAGLE) * with small offset
          MOV  B,A
          MOV  A,THO
          SUBB A,#=(MAX_PERIOD+FINAGLE)
          XCH  A,B                     * get MSB in B, LSB in A
?gpdone
          RET
```

Indirect (or inverse) frequency measurement requires only a single cycle but depends on a faster timebase. The frequency of a signal is the reciprocal of its period, so we can find the frequency very quickly.

The audio frequencies we encounter lie between about 300 Hz and 3 kHz, with periods between 3.3 ms and 333 µs. Measuring those frequencies within ±10 Hz requires a l-us timer clock.

The 89C1051, like most 8051 derivatives, runs its instructions and timers at one-twelfth the crystal frequency. A 12-MHz crystal gives its sole hardware timer, Timer 0, a particularly appealing rate-one tick per microsecond. The '105 1 runs from DC to 24 MHz-although the one I'm using here has a 12-MHz limit-so you can select the crystal or ceramic resonator to match your requirements.

The schematic in Figure 1 shows the 89C 105 1 microcontroller and the front-panel LED array. Last month, you saw the circuitry that produces the *Envelope and *Signal inputs from the transceiver's audio output. Next month, I'll cover the power supply and circuitry driving 'Power Fail.

Timer 0 has an external control gate, but it lacks hardware that mea-

Listing 2—*If no input transitions occur during the period-measurement code in* Listing 1, *Timer 0 triggers this interrupt handler. Setting the* INT_AUDIO *bit simulates an external edge, releasing the tight loop in* **Listing** *1. The loop finds that Timer 0 is not running and returns a zero period indicating the problem.*

```
     ORG   $000B
TmrOVector
     CLR   ET0                   * disable interrupt
     CLR   TR0                   * shut off timer to flag timeout
     SETB  INT_AUDIO             * fake an external edge
     RETI
```

sures the time between two falling edges. Rather than add an external flip-flop, I put the logic in the firmware shown in Listing 1.

The 'Signal input arrives on pin P3.3, the External Interrupt 1 input. As with all 8051 variants, 89C1051 External Interrupts can be either level- or edge-sensitive. In the latter mode, the hardware sets IE1, triggering an interrupt, when it detects a falling edge on P3.3.

The edge-detection hardware works even when the interrupt is disabled, so the firmware simply tests IE1 to find out whether an edge has occurred. A single-branch instruction spins on that bit and determines when the edge occurred within 2 µs. Although that's not as accurate as I'd like, we'll see later that it's close enough for this application.

Measuring a single cycle requires two edge detections-one to start the timer and the next to stop it. However, IE1 provides no information on how long ago the edge occurred, so the code must synchronize on a zero edge before beginning the measurement.

The audio tone may vanish while we are measuring it, leaving the firmware hanging in a single-instruction loop until the tone returns. Needless to say, the measured period has no relation to the actual tone!

Timer 0 can both measure the input period and save us from permanent lockup by counting a value larger than the longest expected period. The code in Listing 1 sets the counter to MAX_PERIOD, a value corresponding to 300 Hz, then enables the Timer 0 interrupt. If the input cycles occur normally, the code shuts off the timer before it causes an interrupt.

However, if no input edges occur, the timer interrupt invokes the code in Listing 2. The interrupt handler shuts off the timer to indicate that the time is invalid and fakes an external interrupt by setting the INT_AUDIO bit. The 8051 hardware allows that sort of trickery because the interrupt flag bits look just like any other Special Function Register (SFR) bits.

That turns out to be a nice way to test your interrupt handlers in a simulator. You can manually set the same flag that the hardware will use, then trace through the handler to be sure it does all the right things.

After stopping Timer 0 at the end of a complete cycle, the code in Listing 1 converts the timer value into elapsed counts by subtracting the starting value. The FINAGLE constant compensates for the average delay between detecting the edges and toggling the TR0 bit.

The value returned in B:A is the period of the cycle, measured in Timer 0 counts. If the input frequency was lower than 300 Hz, making the period longer than 3.3 ms, the return value is zero.

CALCULATING FREQUENCIES

The equation converting a period in seconds into a frequency in hertz is simple:

$$\text{Frequency} = \frac{1}{\text{period}}$$

However, Zerobeat measures the period in microseconds. That formula produces a frequency in megahertz, which isn't quite what we want. The 700-Hz reading lies very, very close to 0 MHz.

Converting from microseconds to hertz requires a numerator scale factor:

$$\text{Frequency(in Hz)} = \frac{10^6}{\text{period (in } \mu\text{s)}}$$

That looks better, until suddenly you realize that 10^6 in hex notation is F4240. That's 24 bits worth of numerator above two bytes of denominator.

In an embedded PC, this is trivial. Put the period in BX, load EDX with 000F, EAX with 4240, then execute DIV BX to get the frequency in AX. Regrettably, we're not using an 80x86 CPU. The 89C1051 is a simple 8-bit microcontroller that unaccountably lacks a 32-bit DIV.

That might be enough to convince you to abandon assembly language for the entire project. A C compiler supporting long arithmetic operations certainly looks tempting, if we weren't confined to the 89C1051's 1-KB flash memory. You can fit a C program into 1 KB, but it better be simple and not drag in much of the standard library.

The good news: there's no reason to let the entire C camel into the tent, when its nose suffices. I wrote Zerobeat in pure assembler by adapting the 32-bit division routine from the Micro-C run-time library.

This may not be quite so simple with other compilers. You'll probably need the library source code to be sure

Listing 3—LongRegA *and* LongRegB, *two four-byte variables in the 89C1051's internal RAM, hold the numerator and denominator used by* LongDiv. *All values use Little-Endian byte ordering, with the least significant byte in the lowest address.*

```
   MOV      R0,#LongRegB              clear the input period buffer
   ACALL    ClearLong
   ACALL    GetPeriod                 measure period
   MOV      LongRegB,A              * save in long register
   MOV      LongRegB+1,B
   MOV      LongRegA,#$40             num for 12 MHz = 1.000
   MOV      LongRegA+1,#$42           scaled by 1E6 to get Hz from µs
   MOV      LongRegA+2,#$0F
   MOV      LongRegA+3,#$00
   MOV      R0,#LongRegA              aim R0 at scale factor
   MOV      R1,#LongRegB              aim R1 at period
   ACALL    LongDiv                   get frequency at [R0]
```

of what you're doing. Some library functions require such extensive setup that they're not practical, but it's certainly worth a look for most of them. Those run-time libraries contain many algorithms and techniques that make you think, "Now, why didn't I think of that?"

C functions, even on the 8051, generally expect parameters on the stack and return arguments in the A and B registers. Assembly code can take better advantage of the CPU registers, so my modified version is rather specific to the Zerobeat's memory layout.

Complex 8051 compilers can optimize the daylights out of the final code, passing parameters in the (few) CPU registers, mapping temporary variables from the stack to internal RAM, and in general, leaving executable code with only the slightest resemblance to your source.

If you're not sure what your compiler is doing, here's a good excuse to do a little spelunking. Micro-C, with more limited goals and a much simpler optimizer, leaves the overall outline of your code untouched.

A pair of four-byte variables, LongRegA and LongRegB, hold the numerator and denominator during division and serve as work areas during the rest of the program.

The Micro-C long-division function uses a third four-byte work area. Even with a dozen bytes in those buffers, the Zerobeat code uses only about half of the 64 bytes of internal RAM in the 89C1051.

Listing 3 shows the few lines that measure the period of one cycle and call LongDiv to convert it into a frequency. The BBS files also include the four-byte numerator you need with an 11.0592-MHz crystal. Zerobeat reads JP1 to figure out which crystal you are using.

The main loop averages four measurements before updating the LEDs, thereby reducing the effect of noise on the display. If you are tuning a weak signal, however, the display still dances around as the firmware attempts to make sense of the hash.

This doesn't affect me very much, as I have trouble following weak Morse signals. You can certainly improve the filtering to match your needs.

Listing 4—*This* **routine converts the frequency difference at** *[R0]* **into** *a nine-bit LED* ***pattern*** *in registers* ***B:A.*** *Using a lookup* **table that** *matches* **frequencies with** *LEDs* **simplified tweaking the results** *until* **they** *looked right. A separate routine distributes each of* ***the*** *nine* ***bits to*** *their output pins.*

```
CvtFreq
     MOV   DPTR,#LEDTable        * aim at lookup table in code space
?cvtl
     CLR   A                     * fetch LSB
     MOVC  A,[A+DPTR]
     CLR   C                     * table - input LSB
     SUBB  A, [R0]
     MOV   B,A
     MOV   A,#1                  * fetch MSB
     MOVC  A,[A+DPTR]
     INC   R0
     SUBB  A.[R0]                *  able - input MSB
     DEC   R0
     XCH   A,B                   * result to B:A for checking
     JB    B.7,?cvtneg           * test for negative result
     JNB   PSW.2,?cvtuse         * -neg and -ovfl is table >= input
     SJMP  ?cvtskip              * -neg and ovfl is table < input
?cvtneg
     JB    PSW.2,?cvtuse         * neg and ovfl is table >= input
?cvtskip
     MOV   A,DPL                 * table < input
     ADD   A,#4                  * skip to next entry
     MOV   DPL,A
     JNC   ?cvtl
     INC   DPH
     SJMP  ?cvtl
?cvtuse
     MOV   A,#3                  * fetch LED bits
     MOVC  A,[A+DPTR]            * MSB to B
     MOV   B,A
     MOV   A,#2                  * fetch LED bits
     MOVC  A,[A+DPTR]            * LSB to A
     RET
* Lookup tables for delta frequency to LED outputs
LEDTable
*           Right      C    Left LED orientation
*                  RROYGYORR      . ..and colors
     DRW   -140,%100000000       * show one LED for far-off low
     DRW   -120,%110000000
     DRW   -100,%010000000
     DRW    -80,%011000000
     DRW    -60,%001000000
     DRW    -40,%001100000
     DRW    -20,%000100000
     DRW    -10,%000110000
     DRW     10,%000010000       * spot on: ±10 Hz
     DRW     20,%000011000
     DRW     40,%000001000
     DRW     60,%000001100
     DRW     80,%000000100
     DRW    100,%000000110
     DRW    120,%000000010
     DRW    140,%000000011
     DRW  $7FFF,%000000001       * one LED for far-off high
```

Now, to put that frequency in lights!

DISPLAYING DOTS

The nine front-panel LEDs show the difference between the input frequency and the stored reference. In this case, reporting an exact numeric value isn't as useful as showing how far off you are in which direction. Yes, the LEDs form a digital "analog" display.

The code in Listing 4 converts a frequency difference into an LED pattern. Rather than writing a series of tests that sort out the LEDs, I created a table holding frequency and LED pairs. That reduced the complex logic into a simple, linear, look-up table.

Each table entry contains a two-byte signed value representing the difference between the input signal and the reference frequency. If the current input lies below that value, the lookup code extracts and returns the corresponding LED pattern. The last entry holds 32767—the highest possible positive number-so no frequency can fall off the end of the table.

Using a table saved my bacon because, after I drilled all the panel holes and wired the LEDs, I realized that I'd gotten them exactly backwards! Rather than rewiring the harness, I simply flipped the table entries. Whew!

The tuning direction may look counterintuitive until you remember how the transceiver works.

The Morse code tone is high-pitched when the transceiver is tuned far below the desired frequency. It falls as the tuning approaches the signal.

If this still seems backwards, you can reflip the table entries or wire the LEDs backwards-and, you could put the rubber feet on top of the case.

A separate routine inverts the bits so that a one in the table turns its LED on, then distributes the nine bits to the appropriate output pins.

You can eliminate this function by permuting the bit patterns in the table to match the outputs and complementing each bit.

RELEASE NOTES

I uploaded the Zerobeat source and hex files last month, so there's no need to duplicate them this month. Dave Dunfield graciously allowed me to use his long-division routine. You can get the rest of his C run-time library, along with the Micro-C compiler, from Dunfield Development Systems.

Somewhat to my surprise, I passed the 13-wpm Morse code test during the Raleigh hamfest in April. Unfortunately, I'm running an 80-m Skywire Loop through the trees at a mere 12′, which gives me rock-solid SO signals.

Next month, we'll finish this seemingly simple project off with an in-depth look at the power supply and calibration routine. The reference frequency stored in internal RAM vanishes if the CPU doesn't enter its power-down mode properly-which means there's more to it than just a regulator! ❑

***Ed Nisley** (KE4ZNU), **as Nisley Micro Engineering, makes small computers do amazing things.** He's **also a member** of **Circuit Cellar INK's engineering** staff. You may reach him at ed.nisley@circellar.com or enisley@ibm.net.*

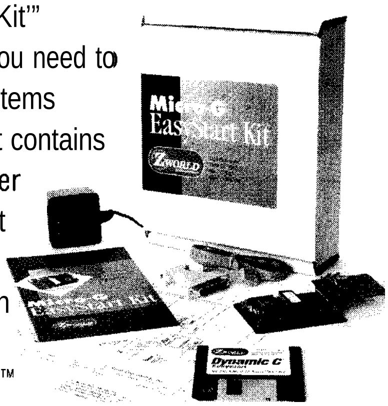
SOURCES

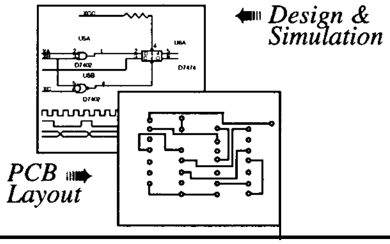
Micro-C compiler, C run-time library
Dunfield Development Systems
P.O. Box 31044
Nepean, ON
Canada K2B 8S8
(613) 256-5820
Fax: (613) 256-5821
http://www.dunfield.com/

89C1051
Atmel Corp.
2125 O'Nel Dr.
San Jose, CA 95131
(408) 441-0311
Fax: (408) 436-4300
http://www.atmel.com/

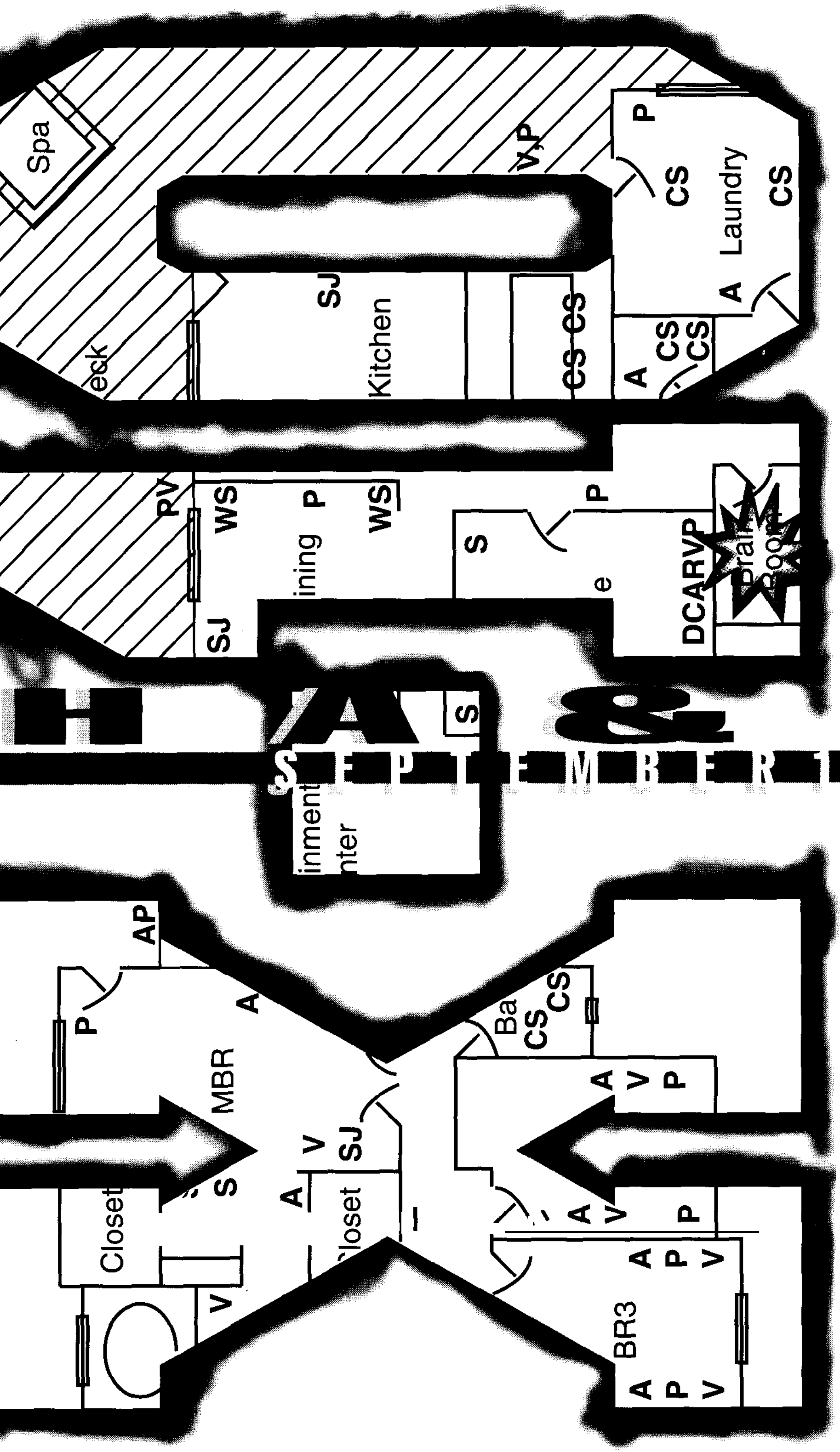
I R S

416 Very Useful
417 Moderately Useful
418 Not Useful





Smart



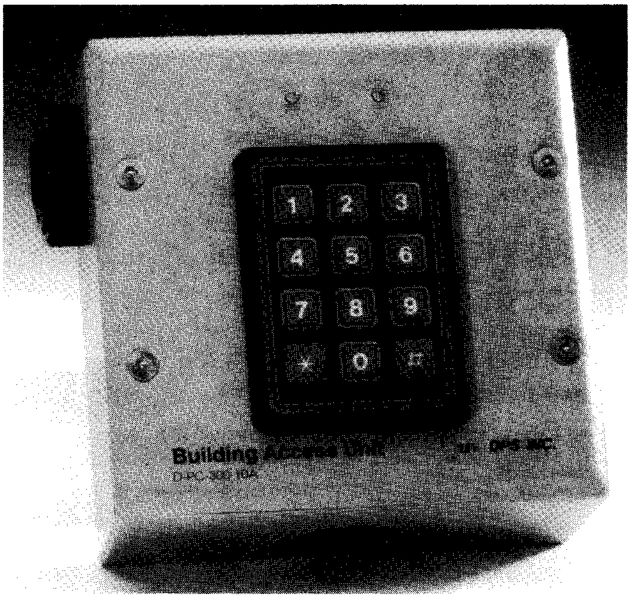
HA&BC

SEPTEMBER 1996

Innovations

IN HOME AUTOMATION & BUILDING CONTROL

edited by
Harv Weiner



BUILDING ACCESS UNIT

Remote, unmanned facilities gain access-control protection with the **Building Access Unit** (BAU). The BAU is a locally programmable, keypad-controlled, building-door monitor with local audible alarm and telemetry-alarm interface.

The unit mounts inside the door and is wired to a door switch. Authorized personnel must enter a keypad code within 30 s or it sounds a 120-db alarm and contacts the facility's telemetry-alarm-system remote.

When an alarm triggers (e.g., Panic or Illegal Entry), an alarm sounds. After 3 min., the alarm stops, but the relay closure to the alarm center remains deenergized. The only exception to this is a special relay Pulse. If the device is in an alarm state and another alarm occurs (e.g., a door opens), the relay energizes and deenergizes, indicating what's occurring on site.

Up to nine personal access codes can be programmed at the keypad. Two contact-closure outputs are provided. The two outputs can be logically wired to the telemetry-alarm remote, providing illegal- and duress-entries and panic alarms. A panic alarm also initiates if the input, output, or power leads to the BAU are cut.

The network version of the BAU transports entry codes to the central alarm master for logging and personnel-location reporting. A stand-alone version of the BAU is also available.

DPS, Inc.
4922 E. Yale Ave.
Fresno, CA 93727
(209) 454-1600
Fax: (209) 454-1688

#510

ELECTRONIC COMMUNICATING THERMOSTATS

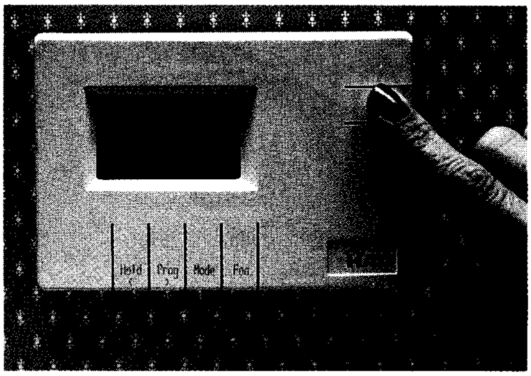
RC-Series Electronic Communicating Thermostats can connect with home-automation systems, utility-control systems, personal computers, burglar alarms, and remote setback switches. Two versions-one for conventional single-stage heating and cooling systems and one for heat pumps with gas or electric auxiliary heat-are available.

The thermostats are programmable for stand-alone or remote operations. They offer flexible 5,1,1 programming with four time periods each for weekdays, Saturday, and Sunday. They also have simple front-panel user controls for Mode, Fan, Set, Hold, and Raise and Lower Temperature. Four communication modes connect to any brand of home-automation or burglar-alarm system.

Utility demand-side-management systems transfer rate data to the thermostat which responds with a customer-selected variance and monitors energy use. RS-232 serial communications to PCs is built in. Up to 127 RC-Series Thermostats can connect to one PC serial port.

Nonvolatile program memory eliminates the need for batteries. Short-cycle protection and random startup after power failure prevents damage to HVAC equipment. Advanced optimum-start technology adjusts heat-pump start time to minimize auxiliary heat (for heat-pump models only). Temperatures are maintained within 1°F for optimum comfort. The easy-to-read LCD display is in Fahrenheit or Celsius and A.M./P.M. or 24-hour time.

Four versions of the thermostats are available in the $200-250 range.



Home Automation, Inc.
2709 Ridgelake Dr. • Metairie, LA 70002
(504) 833-7256 • Fax: (504) 833-7258
sales@homeauto.com • http://www.homeauto.com/

#511

InnoVations

IN HOME AUTOMATION & BUILDING CONTROL

X-10 LIGHTING MODULE

Powerline Control Systems has developed a four-channel, permanently wired lighting module which is X-10 compatible. The LM4 allows convenient control of four individually addressable lighting circuits from just one lighting module. The LM4 is rated at 1000 W per channel with a power not to exceed 2000 W total.

Three versions are available. The **LM4L** controls four triac-driven dimming circuits. The **LM4A** controls four relay-driven nondimming circuits, like fluorescent lights, motors, and fans (0.15 A maximum). The **LM4C** offers two of each.

The LM4 includes several advanced features. Lights brighten from off without being fully on first. `Soft Start` turns lighting on and off gradually.

It also responds to the X-10–defined absolute-level `Preset Dim` command. With this one command, lighting is efficiently set to any one of 32 levels, eliminating the need for multiple `Bright/Dim` commands or knowing the lighting level.

Microstep commands adjust light in 1/200 increments. The brightness level is retained when the module turns off or a power outage occurs. It returns to the previous brightness level and on/off state when power is restored.

The user sets the house and device codes. Program settings are stored in memory which is safe from accidental erasure, including power failures. The user can enable or disable each channel's response to `All Lights On`, `All Lights Off`, and `Soft Start`.

The LM4 is housed in a rugged, all-metal enclosure with a black anodized finish. It mounts to standard single-, double-, triple-, or four-gang electrical junction boxes. LM4 costs **$188** in single quantities.

Powerline Control Systems
9031 Rathburn Ave.
Northridge, CA 91325
(818) 701-9831
Fax: (818) 701-1506

#512

Do-It-Yourself Brain (Room) Surgery

CHRIS ARNDT

If you've wondered how to wire that new house, here's how. Audio, video, automation, telephone, cable, amateur radio-every kind of wire imaginable combines in the central Brain Room.

I've been interested in home automation and A/V signal distribution for over 20 years. I ran speaker wires and video cable, and as soon as the BSR X-10 system came out, I started experimenting with it.

When we built our second home, I remembered the lessons I'd learned. I used the home-run concept and pulled in lots of runs of every type of wire and cable. And, I built an 8' x 9' wire closet.

wiring that might not be available in the future, so I u separate cables for phone, video, speakers, 12-VDC power and antenna feed line for my amateur-radio equipment.

Most homes have the minimum number of AC circuits the National Electric Code allows. Even though our new house has gas appliances and no air conditioning, we have 200-A service and a 42-circuit breaker panel.

Our electrical contractor compared the wiring in our new house to a commercial job rather than a residential one.

WIRE AND CABLE

The phone wire is all six twisted-pair, unshielded, PVC-jacketed cable. The video feeds are a mix of quad-shield or double-shielded RG-6 or RG-59 75-Ω cable. The alarm wire is unjacketed 20-gauge twisted pair.

The 12-VDC feed is PVC-jacketed, 12-gauge, 2-conductor stranded wire. The amateur-radio feed line is either scrounged 50-Ω ½″ Andrew Heliax or ½″ or ¼″ Superflex.

Larger rooms have two separate single-gang boxes for phone and two separate double-gang boxes for audio and video located on opposite walls. No matter how the furniture is arranged, the outlets are only one wall away, as shown in Figure 1.

All the phone boxes have a single home run of six-pair cable. The AN boxes have two home runs of RG-6 per box, and each room has a single 8-conductor 18-gauge home run of audio cable run to one box and looped to the other.

Smaller rooms have a single audio run with a box mounted at wall-switch height, with two 18-gauge pairs running to speaker locations in the ceiling for future in-wall-mount speakers or to single-gang boxes at outlet height for freestanding speakers.

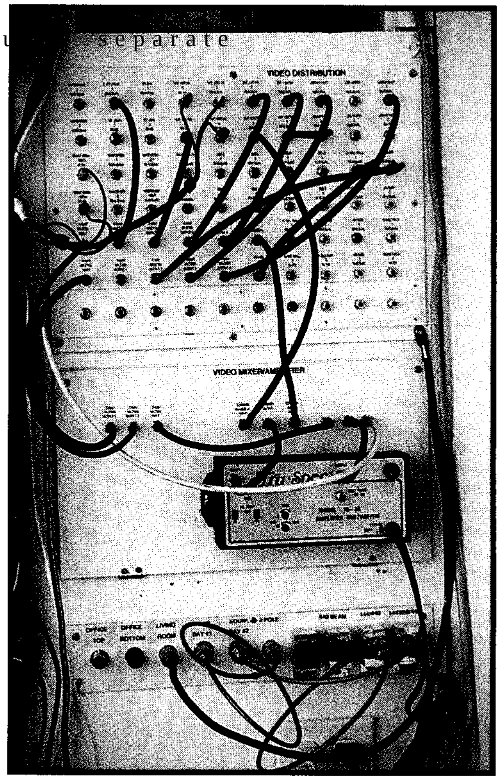


Photo 1: *All the videofeeds terminate on this panel of 70 F bulkhead connectors. At the bottom of the video combining and amplification panel is my amateur-radio distribution.*

The kitchen has in-wall speakers in the vaulted ceiling with a volume control by the sink.

The entertainment center has eight home runs of video cable, one of audio cable, and a TVRO ribbon cable out to the satellite dish. There are also 18-gauge feeds to a subwoofer outlet and two in-wall surround speakers in the dining room.

I also installed runs of audio, video, and phone to the decks, garage, motor-home parking area, and an outbuilding. The total amount of non-AC wiring adds up to about a mile!

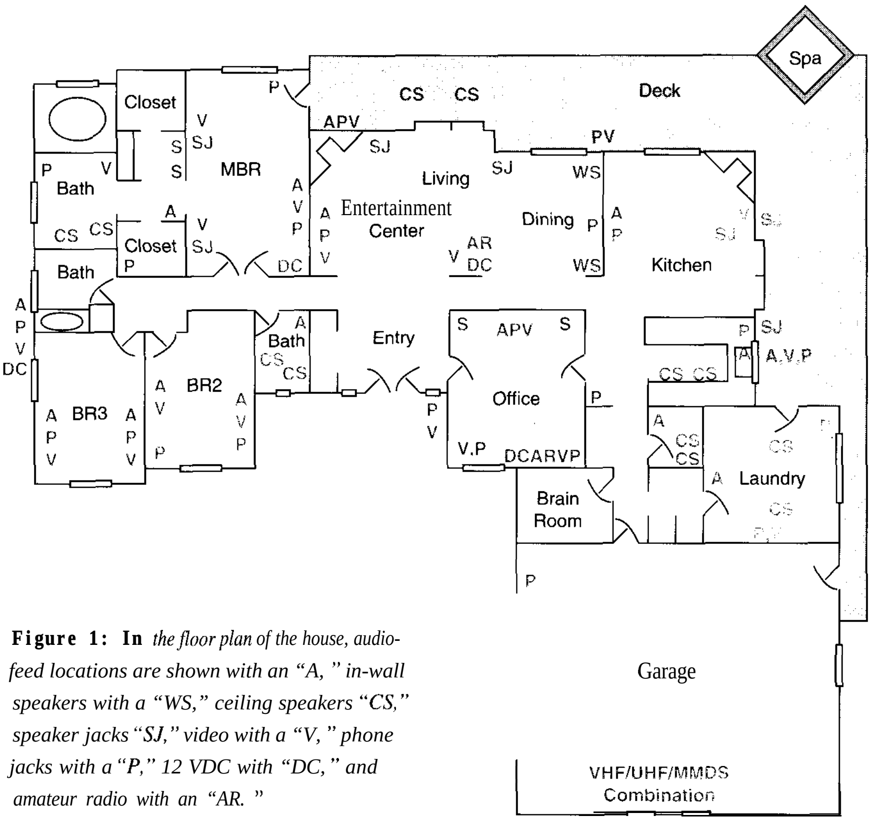


Figure 1: In *the floor plan of the house, audio-feed locations are shown with an "A," in-wall speakers with a "WS," ceiling speakers "CS," speaker jacks "SJ," video with a "V," phone jacks with a "P," 12 VDC with "DC," and amateur radio with an "AR."*

THE BRAIN ROOM

Usually, the room where home runs terminate is called a wire closet. Since ours has so much more in it, my wife christened it "The Brain Room."

I had a real dilemma figuring out what I was going to do with all those wires and cables. I didn't have room to run them all inside a wall. A 10" wire bundle entered the room through a hole in the ceiling centered right in front of the back wall.

After some thought, I built a false wall 16" out from the rear and left walls as shown in Figure 2. I screwed a ledger to the real back wall and ran pine 1 x 2s from it to the false wall forming a wire raceway.

I needed fixed panels to mount equipment, but removable panels for access to hidden cables. I put two 2' access panels at the top and bottom with a 4' fixed equipment panel in the middle.

Most equipment is mounted on the fixed panel. I used some free space behind the top access panel to create small equipment bays for paging amplifiers and such. Larger bays are on the left wall for the X- 10 server, voice-mail computers, printers, and monitors. A small desktop guards the door.

THE EQUIPMENT

The Brain Room's biggest tenant is the phone system and associated equipment. We put in a Panasonic KX-T123211D hybrid telephone system. Hybrid systems use analog paths for the audio, and digital signaling between the proprietary feature-laden telephone sets and the control box or KSU (key service unit).

Our system is capable of up to 12 incoming central office (CO) lines and 32 extensions. It expands to include door phones, door-strike activators, off-premise extensions, and other features.

The KSU has an RS-232 serial port for programming from a computer or printing call details. We use five incoming lines, 31 extensions, a doorphone, and a door-strike adapter. Extensions can be proprietary-feature phones or regular single-line telephone devices.

Phone systems like this are amazingly programmable. CO lines can ring or not on different extensions, and outgoing calls can be allowed or not on the CO lines and extensions. A myriad of parameters can be set—all different for day and night, if you want.

Only three incoming CO lines are hardware telephone-company lines. Another is connected to a cellular-phone adapter that fully emulates a phone line with dial tone and ring voltage.

Of the 3 1 extensions, 21 are for telephone sets. Some extensions are only used when we need them (e.g., the jacks on the deck). Others have regular single-line phones on them.

Eleven are Panasonic KX-T 7 130 feature phones. The 7 130s have LCD displays, speaker phones, and programmable buttons for selecting CO lines, other extensions, or speed-dialing calls.

One of the best things about a home phone system is connecting those telephone-type devices that want to be the only device on the line. We have a four-port voice-mail box, an amateur-radio phone patch, an alarm dialer, an X- 10 voice-response server, a Remote Access Unit (RAU), and a pair of animatronic teddy-bear speaker phones all connected to different extensions.

PHONE SYSTEM TRICKS

The incoming fax line is directed through a three-way switch, which switches on fax tones or DTMF codes (see Figure 3). An incoming fax switches to the fax machine without going through the Panasonic.

An incoming voice call switches to the phone system through a CO-line port programmed to only ring on the voice-mail box. The caller can leave or check messages without disturbing anyone at home or dial through to another extension.

An incoming caller on the fax line that uses the correct DTMF switching code gets switched to the RAU. This box connects to a

Figure 2: *The Brain Room false walls are 16″ out from the back and left walls. The top and bottom plates are screwed to the ceiling and floor joists through the drywall and subfloor.*

phone-system extension. Once the caller validates the DTMF security code, they receive an internal phone-system dial tone and can do anything a regular extension can do.

The 7 130 feature phones can be set up for the speaker phone to autoanswer intercom calls. I use this remote-access feature to listen in at home for smoke alarms or dogs barking. When autoanswering, it beeps rather loudly, so eavesdropping isn't much of an option.

The voice-mail (VM) box is a '386 with a four-port Rhetorex processing card in it and Voice Systems Research voice-mail software. It has autoattendant, voice-mail, and message-forwarding. It integrates fairly well with the phone system and activates a message-waiting light when necessary.

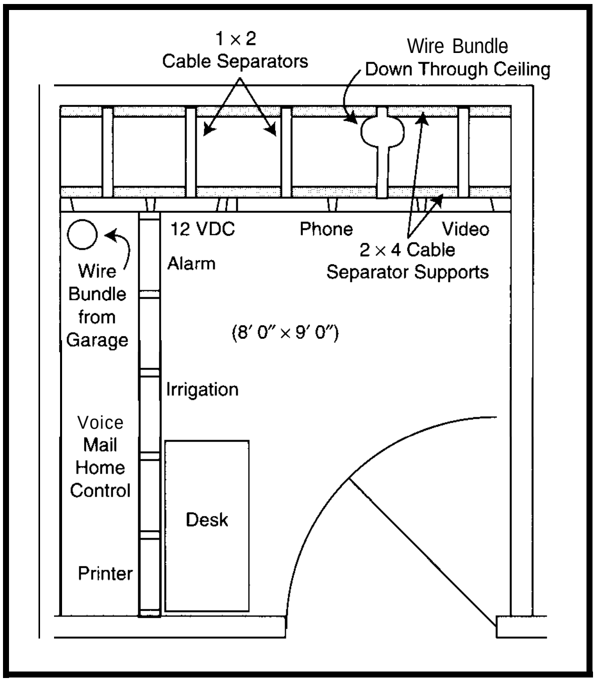
One problem with living in a small college town is that we get a lot of wrong numbers—especially late at night. To combat that, I programmed the KSU and VM box to switch to night mode at 9 P.M. After this time, no phone rings on an incoming call. Everything goes to the autoattendant.

Remember that fifth CO line? We wanted a way for callers to reach us in an emergency. The autoattendant message tells the caller to dial 0 in an emergency.

The VM box forwards 0 or operator calls to extension 129. Rather than plug a phone into 129, I looped it back to CO-line 5. It rings on every telephone in or out of the house. No matter where we are, an emergency call gets to us.

We use the doorphone option rather than a regular doorbell. You can connect up to two doorphones to the KSU.

The doorphone is a small box mounted on a single-gang electrical box. It has a speakerphone inside with a call button. Pressing the call button rings certain phones with a unique triple ring. Answering any ringing phone connects you to the doorphone. If no phone is answered, the ringing stops after three cycles.



The phones have 24 programmable buttons. Twelve buttons intended for CO lines have bicolor LED indicators. Unused ones can be reprogrammed as Direct Station Select (DSS) buttons for other extensions.

When used this way, the LEDs operate as extension-busy indicators. It's great for keeping an eye on phone activity.

I use my old Epson MX-80 parallel interface printer with a serial-to-parallel converter to print call details from the phone system. It prints call duration, CO line used, number called for outgoing calls, and extension used.

X-10 CONTROLS

I've been using X-10 stuff for 14 or 15 years, and I have developed a distributed approach to X-10 control. There is a CP290 computer interface/timer in the Brain Room for timed and security events and a Sundowner control outside for dusk turn-on and driveway motion detection.

The CP290 also operates the low-speed spa pump and heater twice a day to keep the spa filtered and warm. For safety, I programmed the CP290 with multiple off events for most modules. Anything critical or power hungry, like the spa, has a couple of off events scheduled a few minutes apart to deal with inadvertent X-10 signal collisions.

The spa high-speed pump, jet pump, and air blower are controlled at the tub via air switches made for that application. These connect air buttons

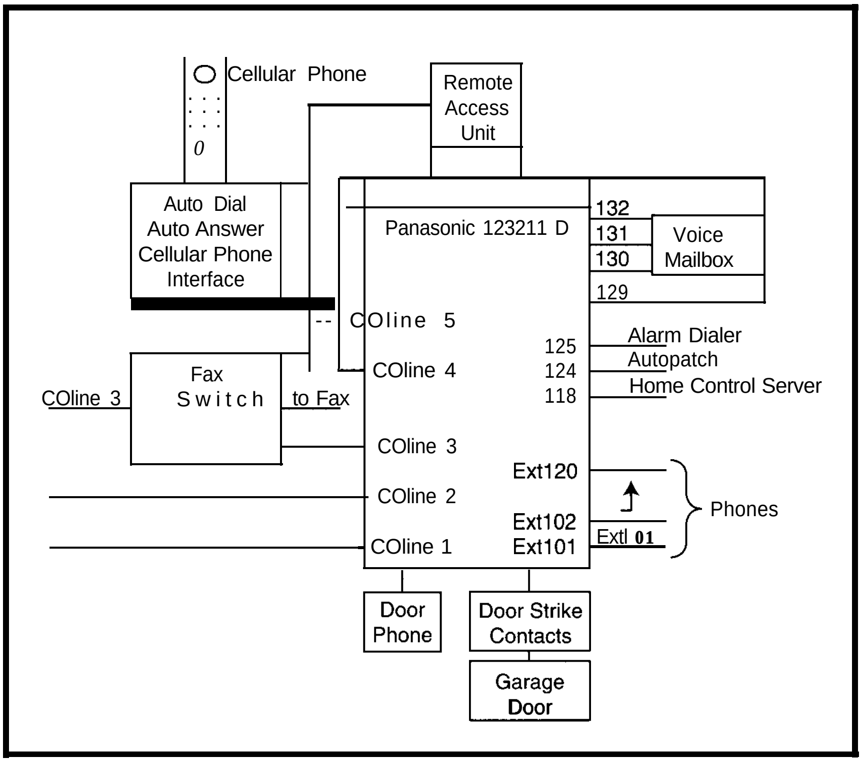


Figure 3: *The phone system is a Panasonic KX T12321 ID. COline 3 is routed through a fax/modem switch and a Remote Access Unit to an extension. One extension is routed back to a CO-line jack for the house phone.*

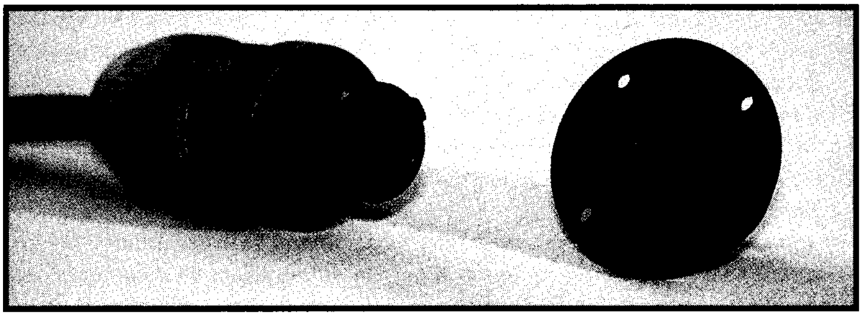


Photo 2: *Neutrik connectors are used to connect tabletop volume control boxes to the wall. These are 8-pole connectors, and all of the contacts are concealed.*

at the tub to air-operated switches in the control cabinet with small-diameter plastic tubing. This keeps electricity a safe distance from the spa.

I built the X-10 server out of an Apple II, AppleCat modem card with the firmware ROM, DTMF receiver chip, X-10 interface, and Micromint Lis'ner 1000 speech synthesizer. It connects to a phone extension.

When you dial the server, the modem answers and asks for a device. I use three-digit codes for each device, so I can use more than one house code.

YOUR OWN CABLE OPERATOR

We're not on wired cable. We get seven channels of wireless microwave MMDS (multipoint microwave distribution system) cable, three channels of VHF off-the-air TV, and satellite reception off our dish.

I managed to combine all these in addition to three channels from a UHF modulator into a single home-cable signal. I started out with the three VHF channels received on outdoor antennas, amplified the signal, and split it out to three TVs.

Then we got wireless cable. The signals come out of the downconverter on cable channels 40-53. So, adding them to the VHF channels was as easy as using a two-way splitter backwards as a combiner, amplifying the combined signal, and splitting it out to the sets.

Then we got our BUD (big ugly dish). The satellite receiver is in the entertainment center with the TV and VCR, so watching it is like watching TV. But, I wanted to watch it on TVs in other rooms.

The satellite receiver has a stereo-encoded channel-3 or -4 VHF modulator. I set it to channel 4 and sent the signal to the Brain Room over coax. There, I ran it through one port of a special channel-4 combiner with the rest of the signals through the other port.

The only problem was that the channel-4 satellite signal interfered with the channel-3 off-air one, and the modulator was double instead of vestigial sideband. Without the lower sideband stripped off, it interfered with channel 3's upper sideband.

I knew there was a local channel-3 translator on UHF channel 57. Up went a UHF antenna. I found, amplified, and combined the station with the VHF and wireless cable signals before it got to the Brain Room.

But, what a hassle to switch TVs from air to cable just for one channel! I started re-viewing what I knew about cable-TV channel numbering and how our TVs worked.

In cable mode, the TVs remapped the UHF channels to high channel numbers. Poking around with the remote revealed the UHF 57 translator on channel 113.

Now, I had a complete VHF, satellite, wireless cable, UHF home-cable system. With experimentation, I balanced the signals and got the correct amplification. I quit messing around with inexpensive amps and splitters and bought better parts from MCM.

The home-cable combining, amplifying, and distribution panels and amateur-radio distribution panel are shown in Photo 1. All the video cables in the house terminate on this one aluminum panel I custom made.

It has 70 F bulkhead connectors, all labeled with the location of the other end of the cable. With the addition of a couple of four-way splitters, the panel is almost full.

Hooking up sources and destinations is as easy as connecting a patch cable made of a short piece of RG-6 and a couple of F connectors. The room and entertainment-center feeds, incoming combined antenna feeds, video and audio from the front-door camera, and scanner-antenna cable terminate here.

Below this panel is the custom combiner, amplifier, splitter panel shown in Figure 4. I made the connections with patch cables. The combiners and splitters are mounted behind the panels and labeled on the front.

For a while, I had the door-camera video and audio line-level signals patched through to the A/V inputs of the TV I wanted to view it on. I could only do one TV at a time without adding a distribution amp. Switching at the set wasn't always easy, and at one TV, despite all the cable I pulled in, I was running out of cables.

So, I got a three-channel UHF modulator to add the door camera and VCR to the home cable. Our house elevation is high enough that it's hard to find open, interference-free UHF channels for the modulator and still be able to use the outdoor antenna.

An interfering signal, unnoticeable on the TV without the modulator, can still interfere with a modulator signal. Also, as weather affects TV propagation around the state, formerly usable UHF modulator channels often go bad.

AUDIO SYSTEM

In our first house, we had discrete wiring home run to the crawl space for two stereo programs. 1 custom built all the wall-control plates and oak table-control boxes with 1 O-W stereo L-pads, A/B program switches, and headphone jacks.

There were two stereo receivers, one for each program. I listened to FM on one, and my wife had TV audio on the other. We knew the limitations of 8-Q speakers in parallel,

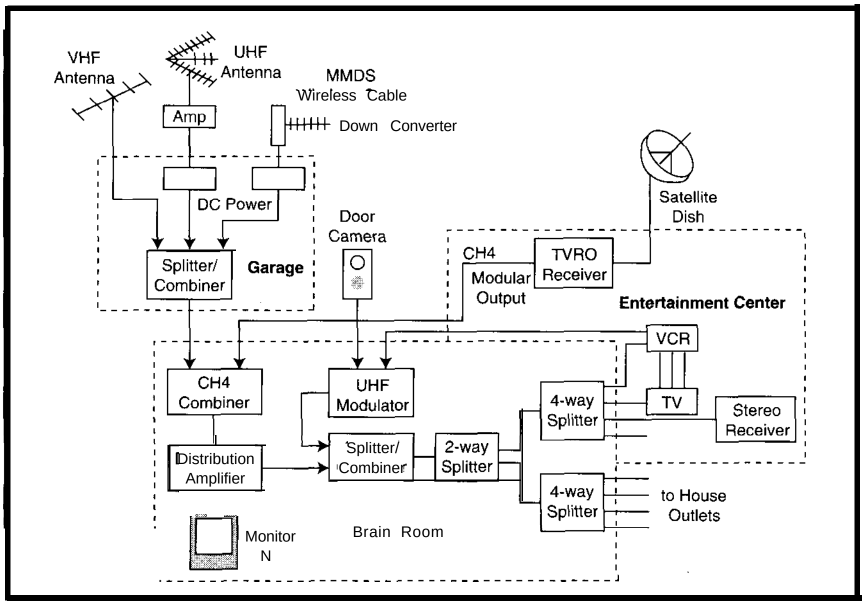


Figure 4: *The home-cable system off-air signals combine in the garage and are sent via one cable to the Brain Room. There, the TVRO modulator output and the UHF modulator@ the VCR and door camera are added before being split into individual rooms.*

and I was always careful of how many speaker sets we had on at once.

I used the same wiring setup in our new house, except for the controls. Russound makes a stereo autoformer volume control called Ultramatch that's jumper selectable for the number of controls paralleled across an 8- or 4-Q amplifier.

In our first house, I used Y-pin Molex connectors to connect the tabletop program selector/volume control boxes to the wall plates. Cutting the rectangular holes in the Bakelite wall plates was difficult and didn't look great. But, they were inexpensive, available locally, and furniture hid most of the wall plates.

I searched a long time for connectors for the new house. I needed at least eight pins to handle two stereo program sources with a common ground and a common ground speaker pair. Finally, I found the &conductor Neutrik Speakon loudspeaker amplifier connectors shown in Photo 2.

Speakons are a double-coaxial twist lock connector with a secondary locking ring and all electrical contacts hidden. Once connected and locked, they can't be unplugged accidentally.

The female panel jack mounts with four screws in a 1 ¼″ round hole. The male cable connector has an integral cord grip for 5⁄16–¾″ cable, and mating female cable-mount jacks are available for extension cables.

TO DO LIST

Home automation never ends, especially since technology evolves. I want to get the home-cable system UHF segment ironed out.

I found another speech synthesizer in the back of an *INK* issue and am designing an autodialing paging server to make announcements and warnings from the phone.

I have a solar-panel array partially built for the 12-V system, and we have so much wind here, I'm going to add a small windmill. I'm thinking about a Stamp-based load shifting controller coupled to an inverter to maximize the wind/solar output, using the battery bank as a flywheel.

Too many projects, too little time....

Chris Amdt, KD6DSI, designs, programs, and installs water and waste-water telemetry systems. As Arndt Electronic Services, he operates small water companies. You may reach Chris at carndt@slonet.org.

SOURCES

Telephone & voice-mail equipment
The Mart
899 Presidential, Ste. 110
Richardson, TX 75081
(214) 238-1 133
Fax: (214) 238-1 132

Autoformer volume controls
Russound
5 Forbes Rd.
Newmarket, NH 03857
(603) 659-5 170
Fax: (603) 659-5388

Video amplifiers & splitters
MCM Electronics
650 Congress Park Dr.
Centerville, OH 45459-4072
(5 13) 434-003 1
Fax: (5 13) 434-6959

Door camera
CCTV Corp.
280 Huyler St.
South Hackensack, NJ 07606
(201) 489-9595
Fax: (201) 489-0111

Neutrik Speakon connectors
Allied Electronics
74 10 Pebble Dr.
Fort Worth, TX 76118
(817) 595-3500
Fax: (817) 595-6444

Text-to-speech synthesizer
RC Systems
1609 England Ave.
Everett, WA 98203
(206) 355-3800
Fax: (206) 355-1098

Basic Stamp II
Parallax, Inc.
3805 Atherton Rd., Ste. 102
Rocklin, CA 95765
(916) 624-8333
Fax: (9 16) 624-8003

I R S

419 Very Useful
420 Moderately Useful
421 Not Useful

Maximizing X-10

JEFF FISHER

Where others insist X-10 is unacceptable, Jeff relishes in finding solutions. No feedback from X-10. Well, not exactly true if you take the right steps. Too **slow—** again, not if you're careful how you program. Curious how he does it? Check it out. He has some very practical solutions to X-10's lack of reliability.

As an engineer turned salesman, I find one type of customer particularly enjoyable-the "I dare you to sell me something" customer. This person walks in and immediately badmouths the products on display. They're usually well-informed and focus on a product's limitations. They seem to come in just to point them out to me one by one.

I've realized that the antagonistic customer isn't just dropping by to ruin my day. They've done the easy part-identifying the shortcomings of the X- 10 scheme. Now, they want the rest of the story.

In this article, I'm assuming you're an "I dare you to sell me something" customer. I'll cover aspects of the X-10 protocol and products related to large or complex installations. Several X- 10 system configuration choices affect speed, versatility, and reliability, so I focus on efficiently sending X- 10 codes from a whole-house controller.

I discuss the most often mentioned shortcomings of X- 10 and some example source code showing how all the points come together. The code uses the hardware shown in Photo 1 to send and receive X- 10 codes using a PC. These points and more are also implemented in the UCIX whole-house controller.

THE ENIGMA CALLED X-10

It's important to know a few things about the X-10 protocol and products:

- they were designed over 16 years ago
- they were designed for absolute minimum cost to manufacture
- they were not designed for computer control or very large installations
- with the exception of vaporware and expensive hard-wired systems, they're the only game in town
- X-10 is still around and in millions of homes worldwide

X- 10 has done little to address the lack of higher end capabilities in the

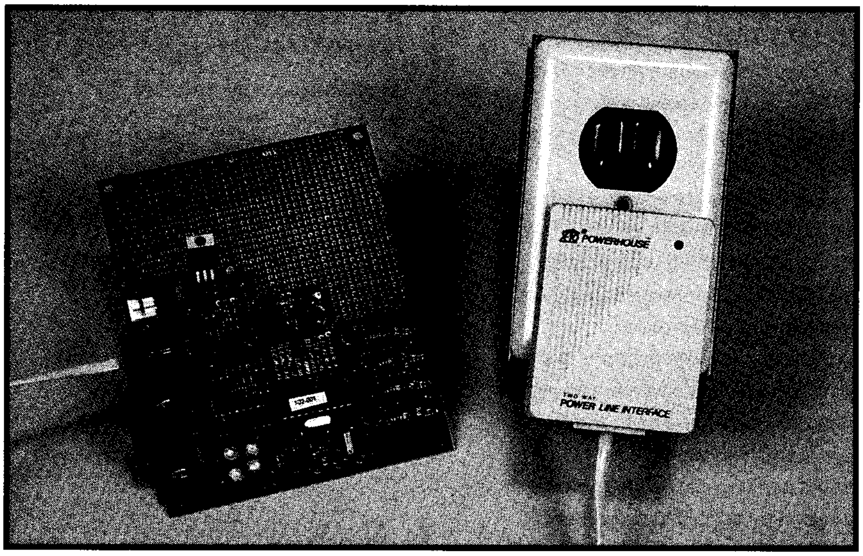


Photo 1 : In *the setup used to develop and test the example software, the PC's serial port connects to the Marrick board (left), which connects to the X-10 TW523 power-line modem (right). The setup lets you easily send and receive X-10 commands.*

protocol. While they've published a protocol specification that includes high-end commands, neither X- 10 nor any licensed manufacturer has implemented them [1]. It's up to us to overcome X-10's limitations.

SEND AND PRAY

The most common complaint about X- 10 is that commands are sent with no confirmation that they were received or acted on. Low signal strength, noise on the line, and collisions can unknowingly keep commands from reaching their destination.

Although this method greatly reduces the cost of transmitters and receivers, it concerns those considering large X-lo-based installations. But, with proper planning, this problem never comes up.

The first step in any significant X-10 installation is to ensure adequate signal strength everywhere [2]. Leviton makes test equipment that measures signal strength as well as bridges and amplifiers that solve low signal-strength problems. With this equipment, you can guarantee adequate signal strength.

Noise is rare, but if you have it, just hunt it down and kill it. Figure 1 shows a line-noise monitor you can make. A signal bridge separates the 120-kHz noise from the power line. A bridge rectifier and capacitor turn the noise into an averaged DC level.

Connect a DVM and track down the noise source. Leviton makes filters to isolate the noise and stop it from coming into your home on the mains.

The problem of collisions is mitigated by several factors. In networking parlance, the original X- 10 system used MA (Multiple Access) technology. Transmitters simply keyed up whenever they needed to send.

When Leviton licensed the X-10 technology for their own product line, they added carrier-sensing circuitry to their transmitters. Now, CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance) greatly reduces the chance of collisions.

The new transmitters wait for the power line to be free of a signal for a length of time dependent on the unit code they are transmitting. If all transmitters have line monitoring and send to different units, a collision occurs only if the line is clear and two transmitter buttons are pressed within 16.7 ms (i.e., one power-line cycle).

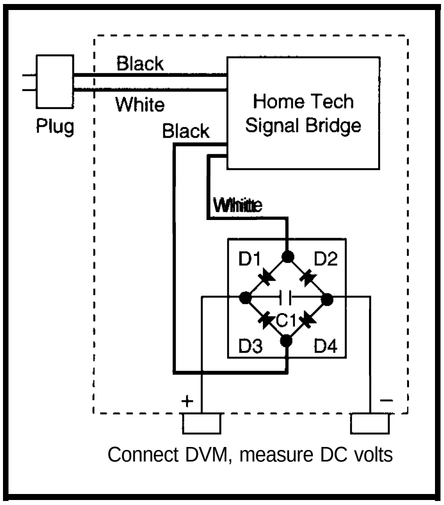


Figure 1: *Build this simple power-line noise-measuring box to measure the relative noise level and track down offending equipment. D1–D4 are 1N4001 or equivalent, and Cl is a 0.2-µF capacitor.*

However, not all Leviton transmitters have line monitoring. No X-10s do. So, further steps are needed to avoid collisions.

One way to reduce collisions is to reduce the amount of time the power line is busy. Later, I'll explain how to reduce the number of codes transmitted, but you can make some initial common-sense plans.

X-10 conveniently transmits events to your whole-house controller. For example, a reed switch or PowerFlash controller tell your whole-house system every time a door opens. But, use X- 10 sparingly in this manner for two reasons.

Many trivial events use up the available bandwidth rapidly. I saw an installation with several dozen interior motion detectors. Each was connected to a PowerFlash that sent an X-10 signal to a home computer.

These motion detectors were designed for the alarm industry and, when triggered, open-close-open-close their contacts. In addition, they retrigger every few seconds.

To show off his new automation system, the owner threw a party and had several dozen guests milling about. When it came time for the big demonstration, nothing worked! The power line was overflowing with dozens of PowerFlashes transmitting nearly constantly.

The PowerFlash doesn't have carrier sense, so if you open the door in the middle of a dimming sequence-which can be several seconds long-you trash the X- 10 codes. Instead, use a Leviton 6314 or 6315 transmitter to send up to eight signals.

If possible, use hard-wired connections to digital inputs in the whole-house controller for this type of signal, thus bypassing the power line completely.

NO TRANSMIT FROM MODULES

The various wall-switch, plug-in, and wire-in modules do not transmit a command when the device they control is turned on or off. If a wall-switch module turns on a light (see Figure 2a), the whole-house controller doesn't know because nothing is transmitted.

Sometimes it's nice to know when someone has turned something on or off. For example, you may automatically want the light turned back off after a delay.

To overcome this limitation, you can use the direct-control configuration shown in Figure 2b. Here, you have the light or appliance wired hot with a wire-in module in or near the device.

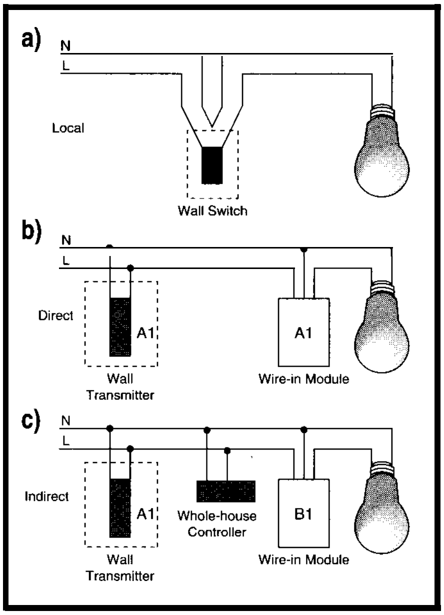
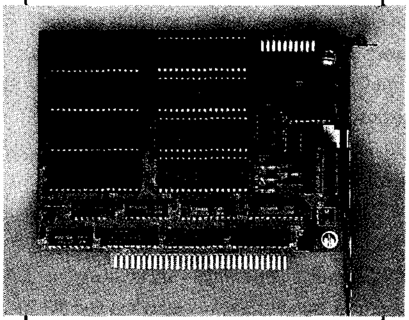
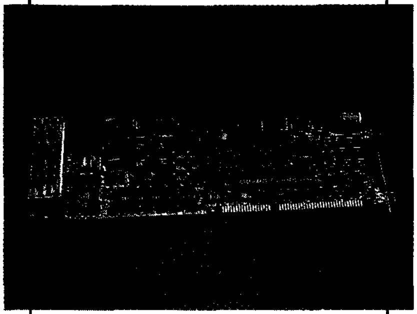
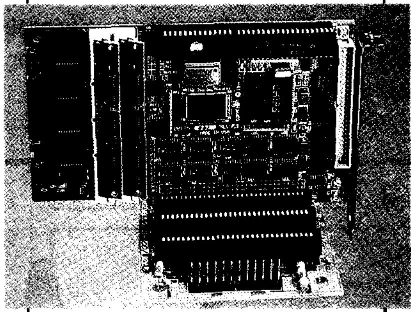


Figure 2: *Of the three remote-control configurations, the local-control configuration is the simplest, but least flexible (a). The direct-control configuration is more flexible, but may require some rewiring (b). The indirect-control configuration provides the greatest flexibility and overcomes several of X-10's inherent limit&ions (c).*







When a user presses a button on a wall-mounted transmitter to send the command, the whole-house controller knows. This configuration requires some wire rearranging, but it doesn't usually require new wires.

NO PRESET DIM

Another complaint about the X- 10 protocol is that you can't gradually brighten a dimmable device (e.g., an incandescent lamp) from Full Off. You first turn the light Full On, then dim it with a potentially long series of Dims.

Figure 3 shows the state diagram for all dimmable X- 10 modules. The only place to go from Full Off is Full On, which is a problem if you want to respond to midnight motion detection with a dim light.

Similarly, you can't set a dimmable module directly to a specific level. If you do this from a whole-house controller, it sends out a series of codes to get the light to the proper setting no matter where it started.

The X- 10 CP290 computer interface solves this by sending the maximum number of bright codes, followed by the desired number of dim codes. Dimming a light from 30% to 20% takes -8 s. Besides blinding you, the power line is tied up longer.

While the X-10 Technical Note lists a Preset Dim command, it is not yet supported by any licensed product.

Note in Figure 3 that Dim never takes a module fully off. If you dim a light all the way down, it appears off, but the module still considers it in a dimmed state called Soft Off. If you send a few Bright commands, it gradually brightens.

Manual-controller users can also use this feature. Rather than turning the hall light out at night, they could dim it all the way down. If they want to get up, they select the unit and press the bright button. However, this approach has three potential problems.

If someone turns the light off the regular way, when the bleary-eyed user brightens the light a little, they'll be surprised.

Another problem arises when someone unacquainted with your dimmed light tries to turn it on from a controller. Nothing happens since the light is already on-just not bright enough to see. The user must press Bright or first Off and then On.

The same thing happens when someone uses the local switch. The first press turns the light off, but since it already appears off, nothing seems to happen. A second press turns it on.

How do we straighten this out? Suppose there's no local switch or control feature (i.e., the direct control configuration.) The whole-house controller listens to X- 10 commands and tracks current states.

Since the whole-house controller knows the device's current state, it sends commands more efficiently. If it knows a light is currently Soft Off, it brightens it a little, waits, and then dims it again.

If the whole-house controller is the only thing controlling the light (i.e., indirect control configuration), it can ensure the light is always dimmed to Soft Off rather than Hard Off.

IT'S TOO SLOW

Another complaint about X-10 is that it's slow. Sending multiple commands to multiple devices can take many seconds.

If the whole-house controller knows the current state of some devices, it may elect to skip some commands entirely. To understand the next level of efficiency, you need to understand how X- 10 command sequences are constructed.

X- 10 codes derive from the original X- 10 transmitter. There are 22 buttons, each transmitting a unique code. Of these, 16 are unit-select codes. The remaining six are command codes that usually operate on the selected unit(s).

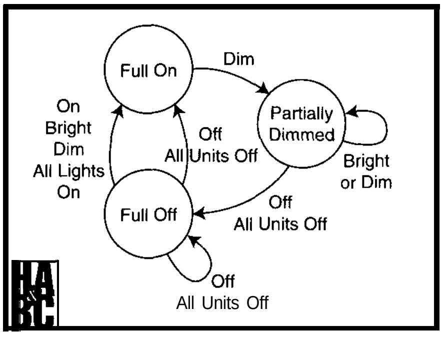
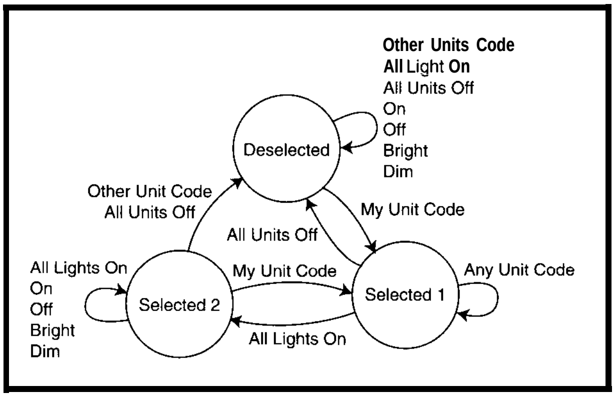


Figure 3: *Dimmahle modules such* us *the lamp module (LM465) cannot go directly from Full Off to a dimmed level. They must go Full* **On** *first.*

Figure 4: *Understanding the selection mechanism is the key to both tracking the state of X-10 modules and compressing X-IO transmissions.*



By pressing more than one unit button before pressing a command, the user gangs units together. For instance, to turn off lights 1, 2, and 6, press 1 2 6 Off. This sends four codes, not the six of 1 Off 2 Off 6 Off.

Figure 4 shows how commands and unit codes select a module. Note that not all modules respond to All Lights On, and modules that cannot dim ignore Bright and Dim.

The last piece of the puzzle is the house code, which is part of every X-10 code transmitted. Modules ignore codes not matching their house code.

A whole-house controller must know two things to track a device—does it respond to All Lights On and is it dimmable? You cannot track the state of devices operated by the local-control feature or a local switch.

If the controller tracks the state, it can intelligently decide to reduce the code count. If the controller wants several lights turned off and knows some are already off, it can send fewer codes. Fortunately for the programmer, most of this rigmarole can be described in tables.

PUTTING IT TOGETHER

Here's an approach implementing these ideas with readily available parts. The demonstration setup uses a TW523, the X-10 Two-Way Interface, to transmit and receive the X-10 codes on the power line [1].

A Marrick X- 10 Interface Board converts the complex, time-critical signals of the TW523 into standard RS-232 data. Each received X- 10 code generates a four-character string. Two- to four-character strings tell the Marrick board to transmit X-10 codes. The demonstration software runs on a PC.

DEVICE DATA

First, we need a place to store information about the devices. An array of 256 structures describes each possible device (16 house codes x 16 unit codes). The device structure contains the device's type and current value.

The device type is a bit field that tells us a few things about the device:

- Bit 0 is True if the device is dimmable. If the device doesn't respond to Bright or Dim, we won't send those commands. It also makes a difference in tracking the device's state.
- Bit 1 is True if we use Soft Control for this device. Instead of issuing Off, dim the device to a Soft-Off state.
- Bit 2 has a True value if the device responds to All Lights On. This is needed to track the state.
- Bit 3 is True if we shouldn't bother tracking the state. (The device has a local switch or uses the local-control feature.)

In practice, only six device types are useful as shown in Table 1.

The structure also contains the device's current value. If the software has just started or if we aren't tracking the state of the device, the value is 12 (i.e., unknown). A value of 11 represents a device that is fully on. Values of 1-10 represent various dimmed levels, with 1 being Soft Off.

The brightening and dimming ramp in a module is analog and varies between modules. The count of 11 steps was derived from testing various modules.

Each step represents two sequential transmissions of Bright or Dim. Since the TW523 only passes through every third received Bright and Dim, this must be compensated for in the software. Some accuracy is lost when tracking a dimmable device, but it's rarely a problem since any move to the extreme self-corrects tracking.

The first step in tracking X-10-module states is monitoring the currently selected modules. The `cur_units` array holds a 16-bit map of the selected units for each house code.

`x10ReadProcess` is called for every X-10 code. It updates the selection maps and uses the `hadcmd` array to track the module selection states as shown in Figure 4.

When a command code-as opposed to a unit code-comes in, pass the command and house codes and the currently selected unit map to `x10Update`. The `x10Change` table updates all selected and tracked device values by indexing into the three-level table with the command, the device type, and the old value.

There are entry points in the code for On, Off, Bright, Dim, Set, Flash, All Lights On, and All Units Off. These high-level commands, depending on the device type, may do different things.

2	Appliance module without local control
5	Soft-control light without local control
6	Nondimmable light without local control
7	Dimmable light without local control
10	Appliance or universal module with local control
15	Dimmable light with local control

Table 1: *A whole-house controller handles an X-IO device more efficiently if it knows a few things about it. Of the 16 possible device types (4 bits of data about each device), these 6 are most useful.*

For example, if you call `x10Off` for a type-5 soft-control device, the software issues just enough Dim commands to take the light from dim to soft off. If you make the same call to a type- 10 device, the software issues an Off X-10 command.

High-level commands call the `x10DoCmd` function with the desired new value for the device. This routine uses the `x10Convert` table to decide how to get the device to the new value.

Index `x10Convert` with the current device value (from `x10-DevTable`) and the desired new value. It returns the command to issue in the low nybble and the number of times to issue it in the high nybble. Bit 3 of the command is set if repetitions of the command are allowed.

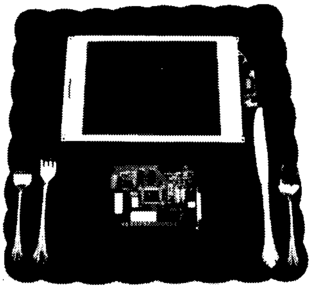
`x10SendCmd` takes the byte from the `x10Convert` table and applies it to the current state. This routine-the most complex and powerful in the sample code-holds off sending any commands until absolutely necessary.

It allows multiple commands such as Bright and Dim to stack up. If it decides that the new command cannot be integrated into the current command, it calls `x10Flush` to flush out (i.e., send) the old command.

Since commands are deferred, the user-level code must call `x10Flush` at appropriate points such as after a timeout or a pass through the command file is complete.

To see the potential of this software, imagine three lights in a living room. Light 1 is set at level 4, and lights 2 and 3 are at level 5.

The whole-house controller wants them all set to level 6. This software



sends 1 Bright Bright 2 3 Bright. It takes -3 s to change the lighting scene smoothly. A less intelligent whole-house controller might send 1 Bright (X8) Dim(X5) 2 Bright(X8) Dim(X5) 3 Bright(X8) Dim(X5), yielding more than 20 s of (very ugly) X- 10 activity.

The software still handles devices with a local switch or control the old way. It turns them Off, On, or Dim.

YOUR HOME AUTOMATED

Does tracking states really work? Yes, provided the tracked modules aren't manually controlled.

Is it worth it to wire devices for direct or indirect control? Sometimes. If a room has several lights and you want lighting scenes, there are many advantages to rewiring for indirect control. But, it probably isn't worth it for light in the kids' room.

PAY DIRT

Now, back to you the antagonistic customer-why do I enjoy you?

You usually walk out with an armload of equipment. If you're savvy enough to see the shortcomings, you usually appreciate workarounds and are technical enough to use them.

Jeff Fisher has been using, configuring, selling, developing for, and writing about X-IO products for 15 years. He is president of HomeTech Solutions in San Jose, CA. You may reach Jeff at jeff_f@ix.netcom.com.

RELEASE NOTES

Software is available on the Circuit Cellar BBS.

REFERENCES

[1] X-10 Home Controls, Inc., "The X-10 Powerhouse Power Line Interface Model #PL5 13 and Two-Way Power Line Interface Model #TW523," Technical note, Revision 2.4.

[2] J. Fisher, "Troubleshooting PLC—A Complete Guide to Diagnosing and Solving Powerline Carrier Problems," Menage Automation, Inc., 1995.

SOURCES

Marrick Board, TW523, UCIX
HomeTech Solutions
10570 S. De Anza Blvd.
Cupertino, CA 95014
(408) 257-4406
Fax: (408) 257-4389
http://www.gohts.com/

Marrick Board
Marrick Ltd.
P.O. Box 950940
Lake Mary, FL 32795
(407) 323-4467
Fax: (407) 324-1291

TW523
X-10 (USA), Inc.
P.O. Box 420
Closter, NJ 07624
(201) 784-9700
Fax: (201) 784-9464

I R S

422 Very Useful
423 Moderately Useful
424 Not Useful

Nonintrusive Current Monitoring

Part 2: Real-time Energy Profile

Last month, Jeff built toroids to check the current consumed in his home. This month, he puts them to work. Using a spreadsheet, he shares that information with us. Listen up if you also want to know where those kilowatts go.

FROM THE BENCH

Jeff Bachiochi

Are we playing right into Big Brother's hands? It seems that our personal files aren't just open to the highest bidder but to anyone who'll pay for them. I am aghast at the information collected about our "private" lives.

Yet, I don't think twice about portable phones, pocket pagers, and the like. Forget about the government planting microchips into our bodies at birth. They don't need to. They can track our every move already while we pay for the privilege.

"No big deal," you say. "The best they can do now is track our general area via the nearest cell."

Well, you won't have to wait long. The personal GPS isn't far off, and it will completely alter the spy industry. James Bond will no longer recover stolen aircraft. He'll be demoted to hall monitor.

On the other hand, spying closer to home is just what the conservationist ordered. I read in the paper recently that, with the nuclear plants offline in the Northeast this summer, more power than ever will be imported.

The electric companies are worried that the increase in current in the high-voltage lines may increase the length of the wires such that they fall below the minimum legal height restrictions. Talk about filament sag.

Last month, I introduced some circuitry I'm using to monitor the energy profile of my home.

This month, I'll collect data and see what it looks like. And, I'll ask the questions, "Is all this practical? Commercially viable?"

RECAPPING

Every conductor carrying current produces a magnetic field proportional to the current flowing through it. The magnetic field can be gathered within a ferrite toroid if the conductor passes through it.

The conductor acts as a single-turn primary of a transformer. A multiple-turn secondary wound on the same toroid receives an induced current due to the magnetic field.

The turns ratio increases the secondary current sufficiently to create a small but measurable voltage across a load. The output can be changed by varying the secondary's turns ratio or by providing a bit of signal conditioning.

An inexpensive front-end multiplexer made from 4016s allows a single ADC to be shared among all the circuits to be monitored. The controlling microcomputer, a Micromint Domino,

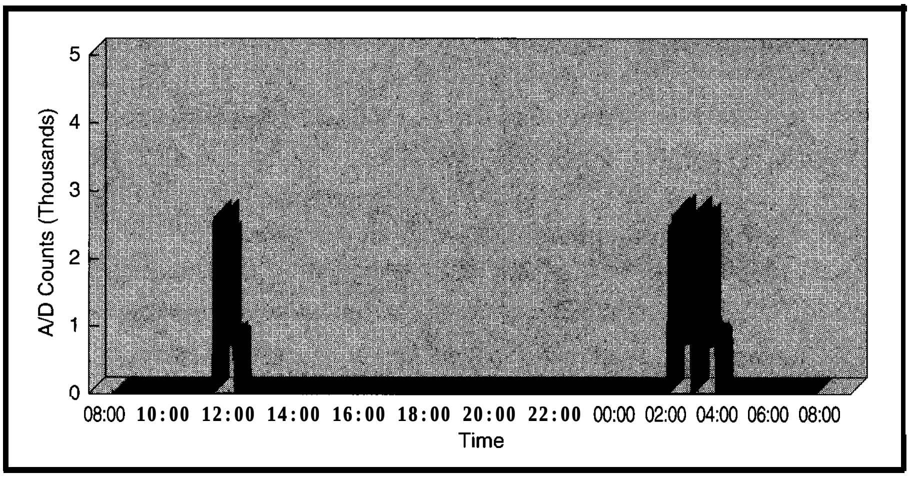


Figure 1--The energy *profile (based on a 20-A circuit) of my Maytag washer shows its use during the day and even for a few washes at night.*

Listing 1--The *BASIC listing in the Domino samples data and displays it graphically or outputs as raw data.*

```
10 DIM V(25)
20  RCLK=1: CLK=2: DAT=4
100  G=GET
110  IF G=O THEN 100
120  FOR X=1 TO 24
130    BAND=255-DAT: BOR=O:  GOSUB 1000:  GOSUB 3000
140  NEXT  X
150 BAND=255-DAT:  BOR=DAT:  GOSUB 1000:  GOSUB 3000
160  FOR X=1 TO 24
170    GOSUB 4000
180    CALL  OFOOOH
190    POP V(X)
200    IF (PORT1.AND.80H)=80H THEN GOSUB 5000
220    BAND=255-DAT: BOR=O: GOSUB 1000: GOSUB 3000
230  NEXT  X
240  IF (PORT1.AND.80H)=0H THEN GOSUB 6000
250  GOT0 100
1000 PORT1=(PORT1.AND.BAND).OR.BOR
1010  RETURN
3000 BAND=255-CLK: BOR=CLK:  GOSUB 1000
3010 BAND=255-CLK: BOR=0: GOSUB  1000
3020  RETURN
4000 BAND=255-RCLK: BOR=RCLK:  GOSUB 1000
4010 BAND=255-RCLK: BOR=O:  GOSUB 1000
4020  RETURN
5000  PRINT V(X)
5010  RETURN
6000  PRINT TAB(10),"1 2 3 4 5 6 7 8 9 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2"
6010  PRINT TAB(10),"                  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 "
6020  FOR Y=100 TO 0 STEP -5
6030    PRINT Y,"%",TAB(10),
6050    FOR X=1 TO 24
6060      IF (V(X)<(Y*41)) THEN PRINT". ", ELSE PRINT"* ",
6070    NEXT X: PRINT: NEXT Y
6100  RETURN
```

has an onboard 12-bit A/D converter. It can multiplex through all the channels using three digital I/O pins for a shift register.

With this configuration, I can include additional ADC channels without a hardware redesign. Now, I can monitor the currents in all branches of my home.

REAL-TIME ENERGY-PROFILE DISPLAY

Circuit current conversions are processed easily with Domino's masked floating-point BASIC. Formatted Print statements let the serial RS-232A output display the house's power profile on any dumb terminal in real-time. Listing 1 is a short BASIC program which autoexecutes on powerup of the Domino and its associated circuitry.

This program has two output modes. The first, featured in Photo 3 of Part 1 (INK 73), is a real-time display of the present energy profile as output to a dumb terminal. This serial output makes good use of the formatted printing available to Domino. The screen updates continuously, showing each of the 24 (21 in use) house circuits and the percentage of maximum current measured in each branch.

The second alternate output is simply a serial dump of the 24 samples whenever requested by the receipt of any character via the serial port. This output is used for data collection.

DATA-LOGGING PROFILE

My 8088 laptop makes a fine data-collection tool. Listing 2 shows a GW-BASIC program written to monitor time and take a sample every 10 s.

The six samples taken every minute are averaged before logging all 24 (averaged) data points to a record on the floppy-disk file. Between samples, a similar real-time display shows the present use of all 24 possible circuits. It updates every 10 s.

This feature is great for testing the hardware. I can turn on the television or other appliance and see its load directly on the laptop's screen.

If you have 24 samples logged 60 times an hour for 24 hours a day, you get a -70-KB file per 24 hours. It's easy to import this file into a spreadsheet.

Set the spreadsheet up so that the measured current, the breaker's rating, and the line voltage are converted into watts. Of course, you could also monitor the lines' voltage which would affect the number of watts used. However, since I found the line voltage to remain steady when I measured it at various times of day, I treated it as a constant.

I used Lotus l-2-3 for my spreadsheet. Its ability to view data in many ways makes it a powerful tool. No wonder spreadsheets are so important to number crunchers everywhere.

VIEWING ENERGY PROFILE

To see instant results, I imported a captured data file. Each sample (con-

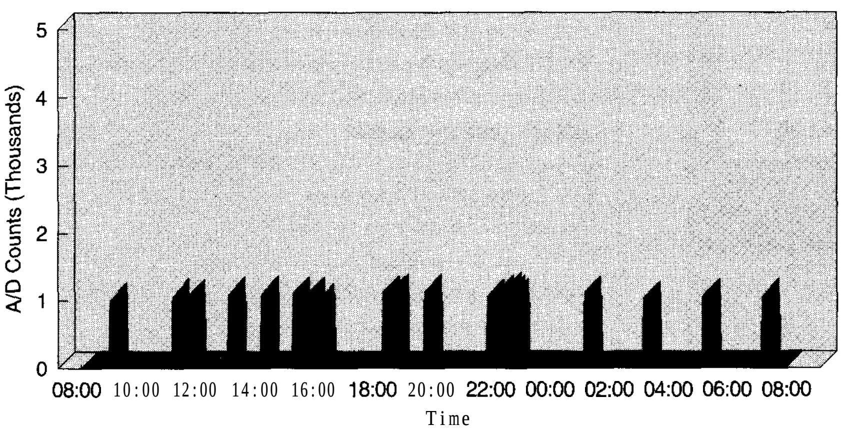


Figure 2—The furnace circuit (an old burner) has short cycles fairly evenly spaced.

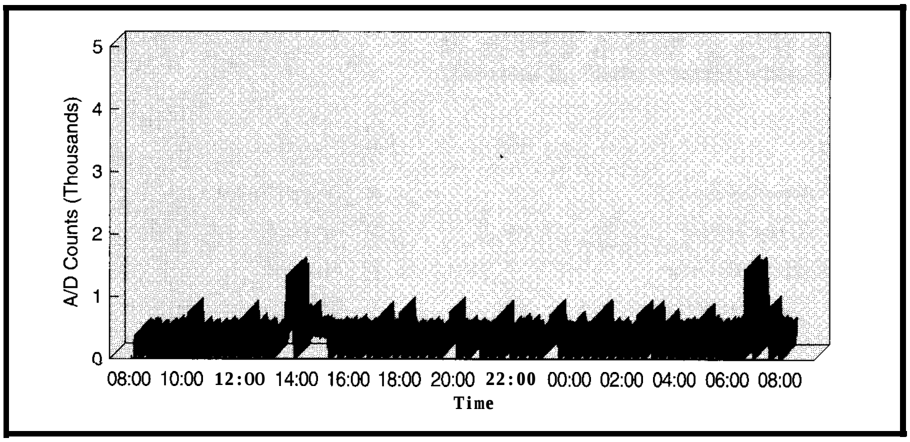


Figure 3-Our fridge, a vertically divided fridge-freezer, is extremely efficient

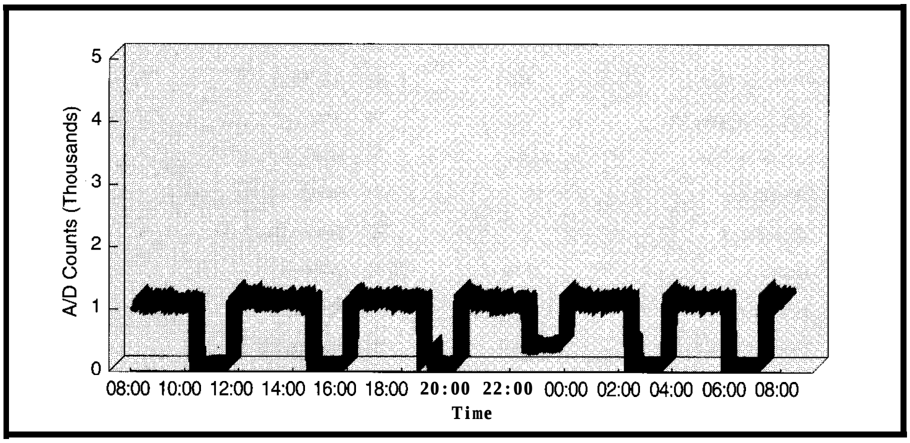


Figure 4—*The* **chest** *freezer in the basement is an old* workhorse *which could be better* insulated.

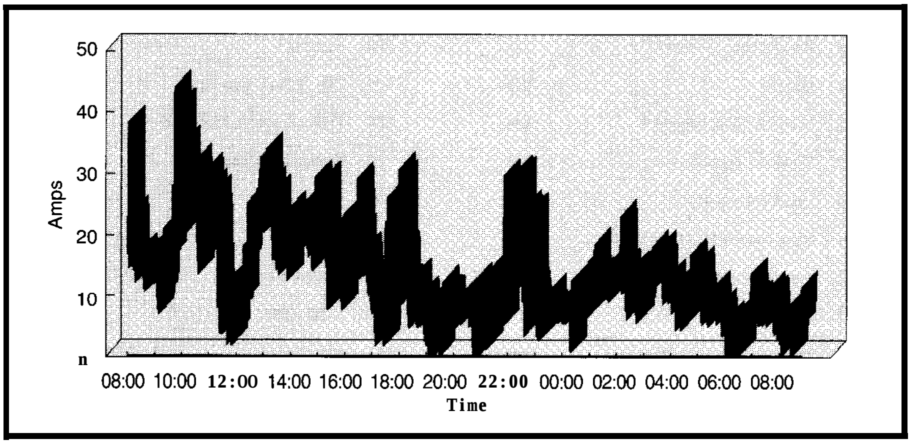


Figure 5—*When I lump all the* circuits together, *a profile of the whole house can be seen.*

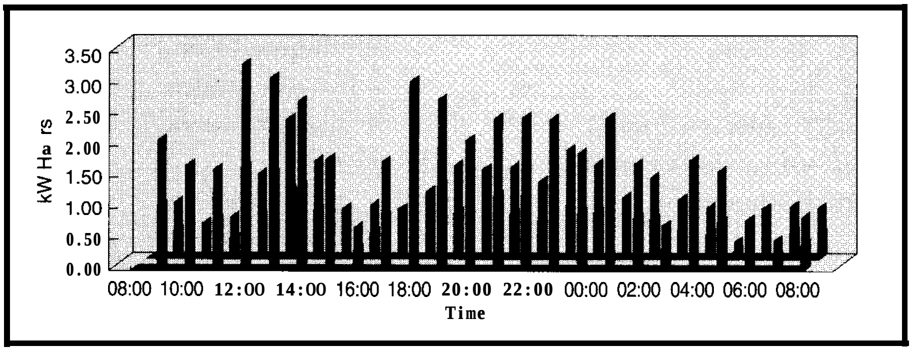


Figure 6—*When I display two consecutive days, the same general profile can be seen each day.*

taining 24 conversion values) has its own row. Columns B-Y represent breaker circuits 1-24.

Column A tracks the hours. Once every 60 rows (minutes), a time stamp indicates when the first sample was taken. Through the graph feature, I can see each circuit's energy profile. Let's look at a few of these to get a feel of what's happening around the house.

Figure 1 shows the profile of the Maytag washer which, according to conversion, runs about 2000 counts peak on a 20-A circuit (about 10 A or 1200 W). Notice the fluctuation of energy used as the wash cycle is followed by the drain and spin cycles. You can track the longer refill cycle and the final rinse, drain, and spin cycles. These timings vary based on the load size (i.e., the water fill level).

Figure 2 profiles the furnace activity. Since it's summer, the furnace isn't required to heat the house. However, it does provide hot water and baseboard heat.

The energy profile shows how little work the furnace has to do to keep a supply of hot water while there's no demand-it's only a few minutes every two hours. During meal times, laundry, or showers, however, it cycles more often. The electricity at only 4 A (i.e., -500 W) is small. I calculated the actual costs of hot water using the current price of oil.

Figure 3 shows the fridge's energy profile. Here you can see the normal cycling of the compressor. The small spikes indicate the extra compressor time necessary whenever someone opens the door and gazes inside as if there's an interesting television program on inside. The larger blips are probably the frostless feature of the freezer cycling once a day.

We keep a chest freezer in the basement, and I was quite surprised at its profile. Figure 4 shows the freezer's long on and off cycles. I had no idea it had such a long cycle time. I'd be interested to know what temperatures it fluctuates between during its one-hour-off two-hours-on cycle time.

OPERATING COSTS

At this point, the data consists of only A/D conversion counts. I can now

Listing 2—*This GWBASIC program runs on my laptop and stores the raw data to a file and displays the real-time data graphically.*

```
10 DIM C$(24): DIM T(24)
20 SCREEN 0,0: WIDTH 80: KEY OFF: CLS: CLOSE
40 F$="b:ftb74.dat": Y=24
60 INPUT"Which COM port (1/2)?";P
70 IF P=l GOT0 100
80 OPEN"COM2:9600,N,8,1,CS0,DS0" AS #1: P=&H2FC
90 GOT0 110
100 OPEN"COM1:9600,N,8,1,CS0,DS0" AS #1: P=&H3FC
110 OPEN F$ FOR OUTPUT AS 2
120 CLS
130 PRINT "Waiting for the minute": PRINT TIME$
140  IF (VAL(MID$(TIME$,7,2))<>0) THEN LOCATE 1,1: GOT0 130
150 N=-10
160 ON ERROR GOT0 420
170 T$=TIME$: M=VAL(MID$(T$,7,2)): M=10*INT(M/10)
190  IF (M<>N+10) THEN GOT0 160
200 GOSUB 240
210  IF (M=50) THEN GOSUB 390
220 N=M:  IF N=50 THEN N=-10
230 GOT0 160
240 ON ERROR GOT0 420
250 LOCATE 1,1
260 PRINT TIME$,"1 2 3 4 5 6 7 8 9 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2"
270 PRINT #1,CHR$(13)
280 FOR X=1 TO Y: INPUT#1,C$(X): NEXT X
300 PRINT " ","                          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 "
310 FOR Z=100 TO 0 STEP-5
320   PRINT Z;"%",
330   FOR X=1 TO 24
340     IF (VAL(C$(X))>40*Z) THEN PRINT "* "; ELSE PR NT ". "
350   NEXT X: PRINT
360 NEXT Z
370 FOR X=1 TO 24: T(X)=T(X)+VAL(C$(X)): NEXT X
380 RETURN
390 ON ERROR GOT0 420
400 FOR X=1 TO 23: PRINT #2,STR$(INT(T(X)/6));",";: T X)=0: NEXT X
405 T(24)=0: PRINT #2,STR$(INT(T(24)/6))
410 RETURN
420 PRINT "Error #";ERR: PRINT "Error in line #";ERL: STOP
```

convert each count into amperes based on the gains set for each breaker's maximum circuit current.

For 40-A breakers, I use the formula:

$$\frac{\text{breakerrating}}{4096} = \frac{\text{current}}{\text{count}}$$

$$\frac{40\ \text{A}}{4096} = \frac{9.77\ \text{mA}}{\text{count}}$$

After calculating the currents for 30-, 20-, and 15-A breakers, I apply these formulas to each column. Then, I sum each row to get the total current used in the house for each minute of the collection period. The profile in Figure 5 shows the current used over a two-day period.

Electricity is sold by the kilowatt-hour. To make the data more meaningful, I convert amperes into watts by multiplying by a line voltage of 120 V.

By averaging 60 samples, I get the kilowatt-hour demand for each hour of the day. This can be extended to calculate the demand in kilowatt-hours for a day or a month.

Figure 6 shows my home's demand profile for two consecutive days. Note how the energy profiles consistently match each time of day. As you can see, our energy demand is about 50 kWh per day. At about 8¢ per kWh, that's $4 per day. Many of us pay more for gasoline to get to work than we do for the cost of electricity.

MORE THAN HOME ECONOMICS

The energy profiles I've seen during this project are a bit surprising. I've learned which appliances cost the most. And, the profile doesn't necessarily follow the day-night cycle I'd constructed in my mind.

While I may follow the "up at 7, down at **11**" profile, other members of my family are night owls. Beverly has a reason. She works second shift, and when she comes home, she needs to unwind. Washing a load of laundry at night makes good use of "off-peak hours." (Way to go, hon!) However, the older boys, Dan and Ryan, often watch a late movie or play video games.

Can this data be used to formulate a strategy for energy conservation?

Since peak demands influence the size of your utility bill, it makes sense to keep the peaks as low as possible by not operating high-wattage appliances at the same time. For instance, don't cook and dry clothes at the same time.

Could the HCS (Circuit Cellar's Home Control System) handle this situation? The HCS certainly could prevent these appliances from being operated at the same time.

Instead of a physical lockout, it makes more sense to notify the person loading the clothes that the oven is in use. Unfortunately, it's still a few years before we see smart appliances.

Prior to beginning this project, I ripped into a circuit breaker hoping to find adequate room to mount toroids internally. But, it isn't possible with today's designs.

So, I challenge breaker manufacturers to consider this feature as a way to keep in step with the future. Conservation is our future.

Let's do it because it's right and before we have no alternative. ❑

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

SOURCES

Domino
Micromint, Inc.
4 Park St.
Vernon, CT 06066
(860) 871-6170
Fax: (860) 872-2204
http://www.micromint.com/

IRS

425 Very Useful
426 Moderately Useful
427 Not Useful

Oh, Say Can USB?

SILICON UPDATE

Tom Cantrell

> Even if you have long hair, the worst snarls are frequently not on your head, but on your desk—at least according to Tom. He applauds industry moves to sort and simplify his desktop wires. He's all for USB.

“If technology is developing at such a rate, why am I still fumbling in that rat's nest of cables behind my PC?”

Testimony to the relevance of that question is the fact I wrote it four years ago! Sad to say, the rat's nest is still there. Serial, parallel, keyboard, mouse, video, audio, and other (I think) cables weave a wicked web.

Since being on all fours started to hurt, my approach with PCs has been to string a single serial cable lifeline to the desktop and mush all the rest of the cables into a dark corner.

The earlier article (INK 28) looked at the ACCESS.bus proposed solution to cabling chaos. Built on Philips 1°C serial-bus technology, ACCESS.bus consolidated the connection of a variety of low- and medium-speed gadgets (100–400 kbps) using a simple and low-cost interface. Witnessing the overall Mac-like morphing of PCs, it's not surprising that ACCESS.bus could best be characterized as kind of an ADB (Apple Desktop Bus) for PCs.

Perhaps I answered my own question in that article when I said the challenge wasn't so much technical as overcoming the “elephant-like inertia that characterizes the PC market.”

Fact is, ACCESS.bus proponents couldn't get that elephant to budge. They gave a party, invited the right crowd, but nobody danced. Box, chip, cable, connector, and software suppliers were ready to tango, each waiting for the other to lead. With increasing awareness that there's never enough bandwidth, ACCESS.bus ended up being too little technology, too soon.

This time, the proposed solution to the cabling crisis and my aching back comes in the form of the Universal Serial Bus (USB). Will it live up to its name? Or, will I be asking the same questions in another four years?

CHICKEN AND EGG

As ACCESS.bus proved, the chicken-and-egg dilemma associated with force-fitting a new interface into the PC isn't easy. We need an immaculate conception to get things rolling.

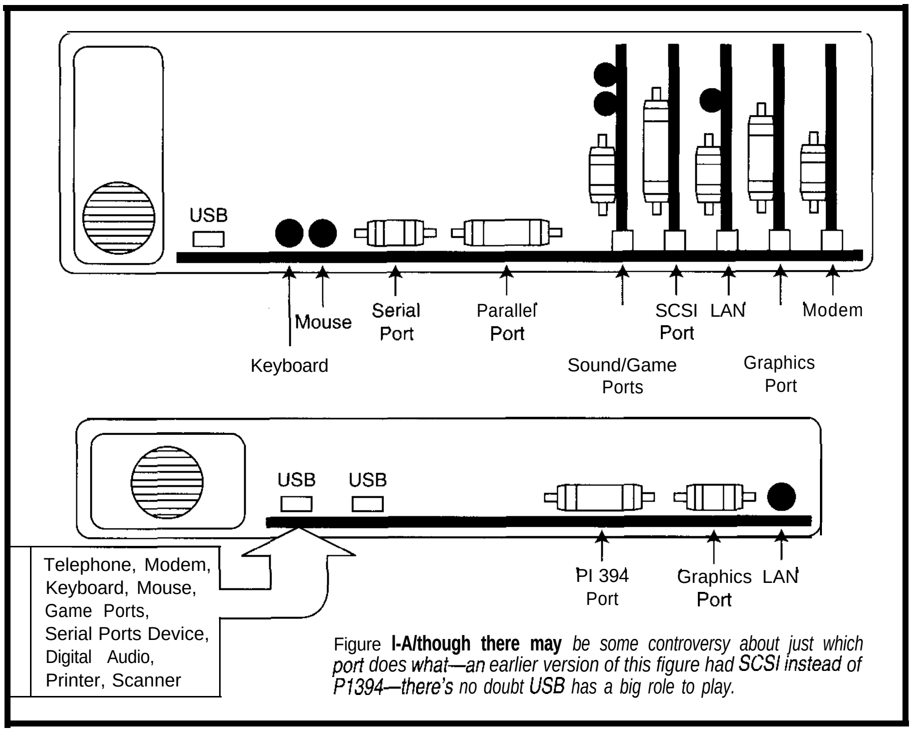


Figure 1—Although there may be some controversy about just which port does what—an earlier version of this figure had SCSI instead of P1394—there's no doubt USB has a big role to play.

In the case of USB, the list of conceivers-Intel, Microsoft, IBM, and Compaq-is certainly immaculate. With a band like that, it's not surprising the dance floor is quickly crowding. The list even includes ACCESS.bus proponents like Philips and DEC.

How to get USB on a motherboard? The answer's pretty simple if you're Intel-just design it in. By now, you know that, besides high-profile CPU chips, Intel is a major OEM supplier of PC motherboards and chip sets.

It won't be long until any system you buy using an Intel motherboard or chip set such as the recently introduced '430HX and 'VX will have USB built in. My guess is, given the balance of power in the PC biz, that means other motherboard and chip-set manufacturers will ignore USB at their peril.

Voilà, the Trojan Horse approach means it won't be long before there's a big installed base of USB-enabled PCs without any messy evangelizing. With millions of empty USB sockets, expect everyone else involved (i.e., cable, connector, device, and software suppliers) to ante up.

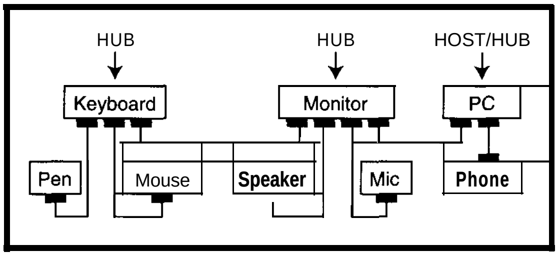


Figure 2—*USB is logically a bus, but it uses a tiered-star topology consisting of a host PC (i.e., the root* **hub), hubs,** *and* **devices.**

THE FOREST

Specifications, data books, user manuals, CD-ROMs, web pages, and VHDL listings-there's no shortage of technical details on USB. I hope USB makes my back feel better, since it isn't helping my eyesight. Before getting into the bits and bytes, let's take a look at the big picture and see what USB means to the average PC user.

Figure **1** shows a scenario in which all the medium- and low-speed I/O is swept into a single USB connector on tomorrow's PC. Actually, there's an intermediate step-now, that is-in which the PC has both the old I/O ports and USB.

There may be some controversy about just what other ports future PCs have and which device plugs where. No doubt, though, USB has a big role to play.

The major advantages of USB compared to the earlier ADB and ACCESS. bus include much higher speed (12 Mbps) and provision for guaranteed on-time data delivery (i.e., isochronous).

Both features are especially critical for audio (i.e., computer telephony integration applications). Forget the dozen or so device limitations of earlier buses. With bandwidth to spare, USB expands the number of connections to a whopping 127.

USB also devotes a lot of effort to power management at both extremes. It guarantees delivery of a lot of power to any device while also offering suspend and resume modes that minimize an individual device's power consumption.

Basically, devices automatically power down when idle and wake up if I/O happens. Besides giving power and taking it away, the specification requires automatic current limiting for each device. Finally, it aims to achieve all of this with low-cost 28-AWG shielded, twisted-pair cable [see Photo 1) and even thinner, cheaper unshielded cable for low-speed (1.5 Mbps) devices.

Lest users feel hemmed in, USB offers a spacious 3 or 5 m (1.5 or 12 Mbps, respectively) between devices. Of course, hot plug and unplug is mandatory because users do it anyway.

I've elaborated all these requirements since they explain the major differences between USB and other desktop buses. Obviously, it's difficult to achieve this with a traditional single-wire bus. Power distribution and control of dozens of devices over hundreds of meters is a showstopper.

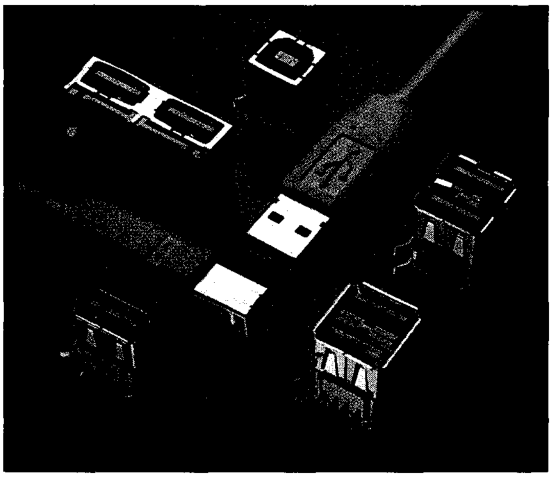


Photo 1-Siemens, *AMP* **(shown** *here),* **and** *others are gearing up to supply USB cables and connectors.*

Instead, USB adopts a tree-like or tiered-star point-to-point topology consisting of hubs (the root hub being the host PC) and devices as shown in Figure 2. This divide-and-conquer approach distributes the problems across the network.

Of course, this approach calls for more silicon than yesterday's simpler schemes. As technology marches on, however, the cost of the extra transistors becomes less noticeable.

It's important to understand that—despite the tiered-star wiring-USB is still logically a bus. In normal operation, hubs act as repeaters, so each wire segment carries the same information at the same time (ignoring the prop delay at hub).

Autonomous hub activity is largely housekeeping (e.g., power control and hot plug) rather than data transfer. Notably, hubs can be bus or externally powered, the latter option boosting an attached-device allowed power to [who needs CMOS?) 0.5 A. Also, hubs can contain devices, as in a keyboard (hub) with a mouse (device) port.

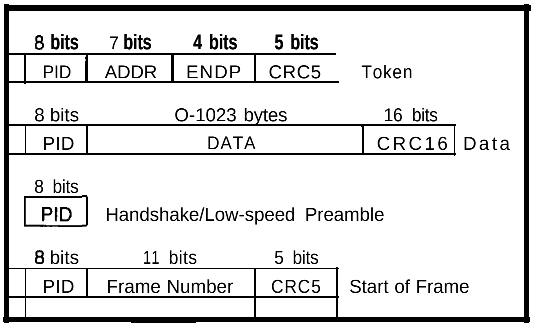


Figure *3-Besides data, USB has special packets for granting media access (i.e., Token), transaction handshake, and signaling the start of a frame.*

In many ways, USB performance and architecture are closer to LAN than to the older desktop buses. However, USB eschews the typical contention-access method in favor of host-PC-controlled access.

It's a polling mechanism in which nobody speaks until spoken to by the PC. Unlike contention methods, host-controlled access has major advantages when it comes to guaranteeing timely service.

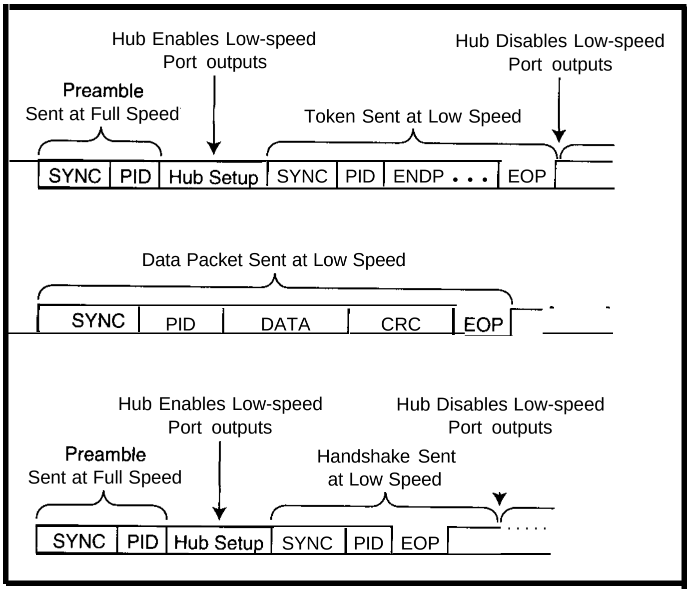


Figure *4-Besides illustrating the basic token-data-handshake transaction sequence, hubs must be able to handle both high- (12-Mbps) and low-speed (1.5-Mbps) transfers.*

THE TREES

USB uses a 1-ms frame of reference. On power-up or when the network configuration changes, the host PC goes through an enumeration process that interrogates each device for a wealth of information including bandwidth and latency requirements.

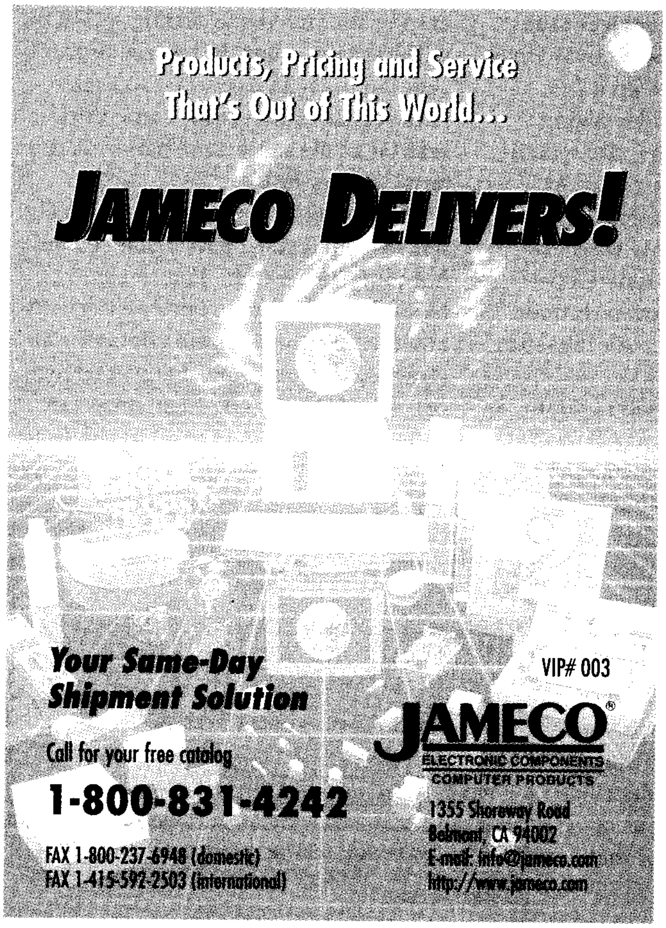
As enumeration proceeds, the host gives each device the required share of the 1 ms. If the full frame is allocated, no other devices can connect. However, once a device is allowed to board the bus, it's guaranteed to find the seat (i.e., the bandwidth) it needs.

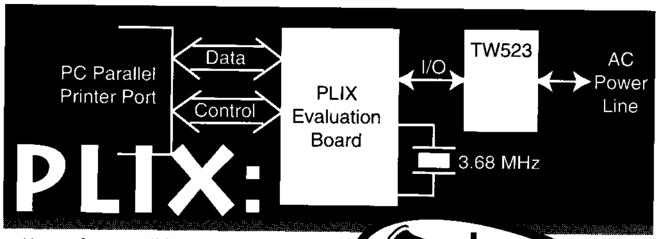
The 1-ms frame thus consists of many packets of various types, including data and control (see Figure 3). At the start of each frame, the host issues a Start of Frame, followed by a token-data-handshake packet sequence for each scheduled device.

The token, which grants media access, specifies a device address and the endpoint specifier. Endpoint 0 is required for every device to establish connection and gather the configuration info.

Other endpoints may be used to accommodate multidata channels [e.g., 8-channel data-acquisition gizmo) or status and control ports. In USB-speak, each endpoint needs its own buffer in the host. The connection between them is called a pipe.

Yes, there's a lot of overhead for small transactions, but 12 Mbps goes a long way. Even in worst case (single-byte packets ≅ 1000% overhead), USB





easily handles the equivalent of keyboard, mouse, and serial and parallel ports without a hitch. Practically speaking, even these lowly devices typically include a few bytes of buffering, which cuts overhead a lot.

The lower-speed (1.5 Mbps) option may be adequate for slower and more cost-sensitive devices like a mouse. As Figure 4 shows, a hub handles rate matching since its upstream connection is always high speed.

Higher-speed devices and improved efficiency call for larger buffers (i.e., packet size) as dictated by device throughput, high- versus low-speed, and 1-ms update rate.

As a simple example, CD-quality audio at 160 kBps calls for 160 bytes every frame. With larger packet size, overhead is cut dramatically. The audio consumes only a small fraction of the available bandwidth (i.e., 12 Mbps = 1.5 kBps).

In any case, packet buffering is mandatory for time-sensitive or isochronous data such as audio. It's better to get a packet on time, even if some bits are broken-which is why CRC error checking and retry mechanisms are disabled for isochronous transfers.

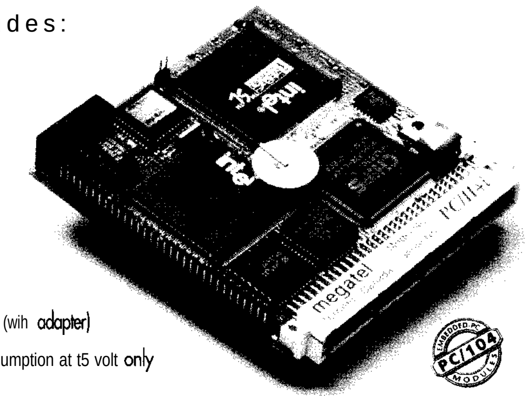
However, as shown in Figure 5, a device can't guarantee exactly where within a frame its packet will appear. Avoiding jitter typically calls for two packet buffers allowing simultaneous fill and dump in a ping-pong fashion.

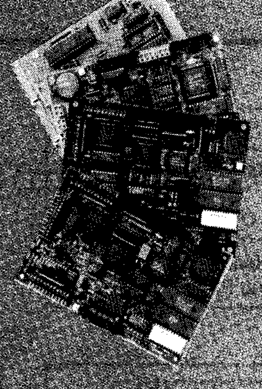
With copious massaging by silicon and software, data eventually gets the boarding call and heads for the wires—a differential, bidirectional, half-duplex pair (D+ and D–).

Electrically, the transceiver specifications are similar to conventional differential line drivers and receivers (e.g., RS-422 and RS-485), although there are possible gotchas to watch for. For example, the rise- and fall-time specs are different for high- and low-speed transfers to avoid EM1 on the latter's unshielded cable.

However, what really makes this a differential pair with a difference is that, besides simply encoding the data (i.e., two states, 1 and 0), it handles additional states.

For example, driving both D+ and D- low produces a Single-Ended Zero



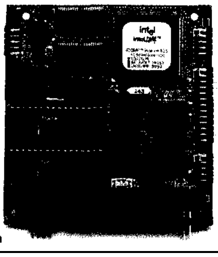


Sets the Pace
in Low Power,
High Performance
PC/I **04Technologies**

100 MHz
486DX4
$995

Features:

486DX4
100 MHz
16K cache
8MB DRAM
SSD
EEPROM
WDT
2 Serial
1 Parallel
PS/2 mouse
Keyboard
Quick Boot
Virtual devices
Flash File System

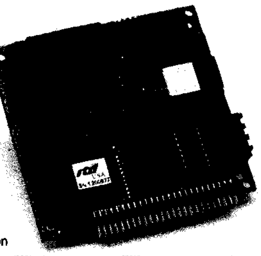


CMV486DX4 Fully Integrated PC-AT with Virtual Device Suppport

200 kHz
12-bit DAS
$525

Features:

200 kHz
12-bit A/D
16SE/8DI
Scan
Burst
Multiburst
1K CGT
Triggers
FIFO buffer
Data marker
12-bit D/A
16-bit DIO
+5V operation



DM5408 200 kHz Analog I/O Module with Channel-Gain Table

Make your selection from:

6 cpuModules™

SuperXT™, 386SX, 486SLC, 486SXLC2, 486DX2 and 486DX4 processors. SSD, 8MB DRAM, RS-232/422/485 serial ports, parallel port, IDE & floppy controllers, Quick Boot, watchdog timer, power management, and digital control. Virtual devices include keyboard, video, floppy, and hard disk.

7utilityModules'"

SVGA CRT & LCD, Ethernet, keypad scanning, PCMCIA, intelligent GPS, IDE hard disk, and floppy.

20 **dataModules®**

12, 14 & 16-bit data acquisition modules with high speed sampling, channel-gain table (CGT), sample buffer, versatile triggers, scan, random burst & multiburst, DMA, 4-20 mA current loop, bit programmable digital I/O, advanced digital interrupt modes, incremental encoder interfaces, opto-isolated digital I/O&signal conditioning, opto-22 compatibility, and power-down. 18-bit voltage to frequency converter module.

Real Time Devices USA
200 Innovation Boulevard • P.O. Box 906
State College, PA 16804-0906 USA
Tel: 1 (814) 234-8087 • Fax: 1 (814) 234-5218
FaxBack®: 1 (814) 235-1260 • BBS: 1 (814) 234-9427
http://www.rtdusa.com/home.htm

RTD Europa
Budapest, Hungary
Tel: (38) 1 325-1130
Fax: (36) 1 328-6737

RTD Scandinavia
Helsinki, Finland
Tel: (358) 0 346-4538
Fax: (358) 0 346-4539

RTD is a founder of the PC/104 Consortium and the world's leading supplier of intelligent ISA DAS interfaces.

#141

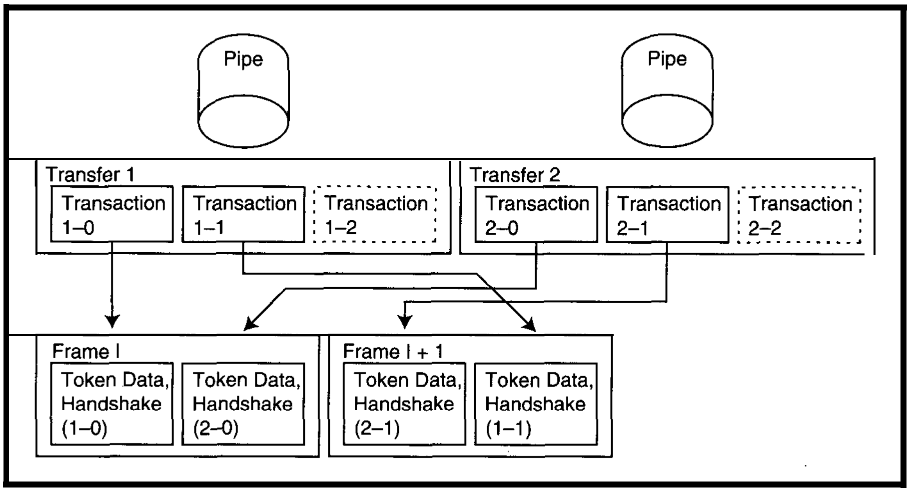


Figure 5—*Although transactions for each device are kept in order, their exact timing can't be determined from frame to frame. So, jitter must be hidden by packet buffers.*

(SEO) **state** that acts as the End of Packet (EOP) identifier. Similarly, the K state acts as Start of Packet (SOP), while J is entered when the bus is idle.

Along with the additional states, a few resistors provide key features such as suspend/resume, hot plug, and high- versus low-speed configuration. As shown in Figure 6, both data lines pull down weakly in a hub port, while a single line pulls up strongly in a device. So, the hub determines when something is connected (i.e., a data line is pulled up) and if so, what speed it prefers (i.e., which line is pulled up).

Ultimately, once you get down to 1s and 0s on the wire, USB adopts conventional techniques.

The data is Non-Return to Zero Inverted (NRZI) modulated which means a transition encodes a **1** and no transition a 0. The clock is shipped along with the data (saving wires and avoiding skew problems) and reconstructed at the receiver with a digital Phase-Locked Loop (PLL).

To allow the PLL to lock up, each frame is preceded by a Sync pattern. Bit-stuffing ensures enough data transitions to maintain synchronization.

THE SEEDS

If you believe the USB harvest will be bountiful, get your seeds in the ground now. The best place to start is with the USB developers' group. You can download the specs from their

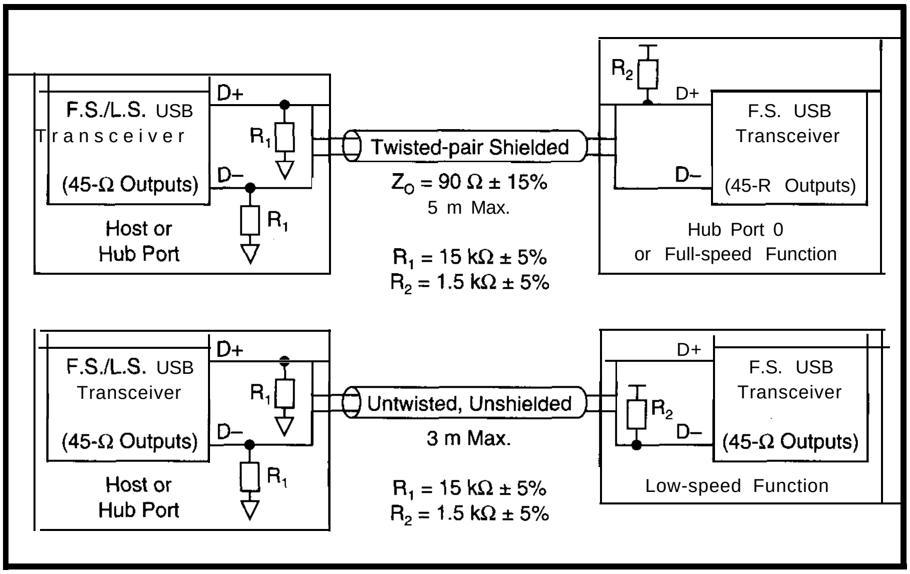


Figure 6—*Resistor configuration allows the host to determine whether a device is connected and at what speed. Notice the different cable specifications (i.e., shielding and distance) for each speed. A host or hub must be able to handle either speed connection.*

Web page. They will point you to a variety of resources.

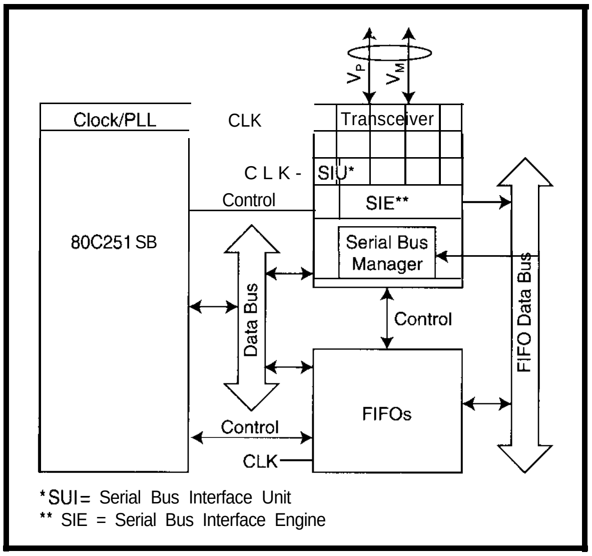


Figure *7-The 82930 combines a high-end d-bit controller-the '251—with USB interface logic.*

Not surprisingly, besides USB motherboards, Intel is the first supplier of device- and hub-type silicon in the form of the 82930A (see Figure 7). It combines their '251 CPU (see "Plan '251 From Outer Space," *INK 56)* with special USB hardware in the form of a serial interface engine and extra FIFOs.

While USB does call for a lot of silicon, the '930A ($6 in 10,000s) is arguably overkill for something like a mouse. Intel says they have plans for lower-cost variants.

I suspect there'll be more sources for silicon soon, but in the meantime, you could roll your own. However, be warned that it's not like simpler desktop buses. You can forget trying to get away with simple bit-banging on a cheap micro.

The best option is to contact the USB folks and find out how to get a copy of the synthesizable VHDL for the serial interface engine (the one used in the '930A) which handles the gnarliest details (e.g., clock recovery, bit-stuffing, CRC, etc.). It boils down to about 3000 gates in your favorite PLD, but that's without FIFOs since their size is best determined on an individual application basis.

Whatever the brains of your USB gizmo, Philips provides the brawn in a specialized USB transceiver (see Figure 8) that meets the rather finicky electrical and timing specifications. The 'P11 runs at 3.3 V, but has 5-V tolerant I/O.

The incoming data is output as a level on RCV, but each signal of the differential pair (VP, VM) is also separately available to decode special states (i.e., VP, VM both low is the SE0 state). Splurge, it's only 35¢ in 10,000s.

When it's time to test your little USB beauty, you'll find many of the tools, chips, and code make the job harder. The problem, ironically, is that all the stuff works too well.

Consider if you want to test various error conditions such as CRC glitches, clock drift, jabbering, and so on. Unfortunately, your USB-enabled PC probably doesn't have any BIOS entries along the lines of "Send An Error-Filled Packet."

To get down and dirty, use tools like the Windows-based test gear from Computer Access Technology. USB Detective ($2950) is a line monitor that deciphers bus activity in great detail. Traffic Generator ($3750) lets you exercise your design with packets of all sorts (including badly broken ones).

BACK TO THE FUTURE

Some may quibble with the technical aspects of USB—whether a particular device is best as a hub, device, or both. Similarly, the low- versus high-speed option may cause trouble if suppliers save a buck with the former and clog the bus.

However, technical challenges pale in comparison to PC market inertia. Despite the players' pedigree, the fact remains that as I write, new retail PCs don't have USB ports.

However, I expect it won't be long before USB-enabled PCs and the first wave of gadgets (keyboards, mice, etc.) appear on the shelves.

Indeed, I'm putting my PC upgrades on hold until I see how USB shakes out. I say, give me freedom from cable chaos or give me death (or at least a Mac). ■

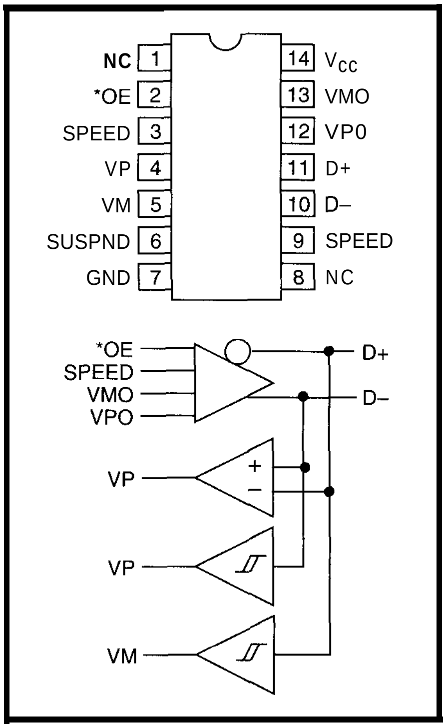


Figure 8—*The PDIUSBP11* transceiver from Philips is a *simple* way to *get* on the bus.

Tom Cantrell has been working on chip, board, and systems design and marketing in Silicon Valley for more than ten years. He may be reached by E-mail at tom.cantrell@circellar.com, by telephone at (510) 657-0264 or fax at (510) 657-5441.

SOURCES

USB Connectors:
AMP
MS 024-010
P.O. Box 3608
Harrisburg, PA 171053608
(800) 522-6752
Fax: (717) 780-4508

PDIUSB611
Philips Semiconductors
811 E. Arques Ave.
Sunnyvale, CA 94088-3409
(408) 991-5207
Fax: (408) 991-3773
http://www.semiconductors.philips.com/

Siemens Connector Division
20-F Commerce Way
Totowa, NJ 07512
(800) 410-6057
Fax: (201) 256-6375
http://www.siemens.de/ec/connectors/usb.htm

USB Implementers Forum
M/S JF2-51
2111 NE 25th Ave.
Hillsboro, OR 97124
(503) 264-0590
Fax: (503) 693-7975
http://www.teleport.com/~usb/

USB Detective, Traffic Generator
Computer Access Technology Corp.
3375 Scott Blvd., Ste. 410
Santa Clara, CA 95054
(408) 727-6600
Fax: (408) 727-6622
http://www.catc.com/

Literature Packet **#L154,** 82930A
Intel Corp.
P.O. Box 7641
Mt. Prospect, IL 60056-7641
(800) 628-8686
http://www.intel.com/embedded/

IRS

428 Very Useful
429 Moderately Useful
430 Not Useful

ENGINEERING PROFESSORS LISTEN UP!

The reason you read *Circuit Cellar INK* is obvious. Like thousands of others, you have come to realize that *Circuit Cellar INK* presents technically relevant computer-based applications in a comprehensive and instructive manner. In fact, our projects are presented just the way you'd do them if only you had the time.

Since 1994, we have provided thousands of college students with industry-quality development tools and information. This program is **FREE** and because of the continued success we have experienced, we'd like to extend our offer to the 1996-1997 school year.

If you're an engineering professor or teach qualified technical students, we'd like to give you and all of your students free subscriptions to *Circuit Cellar INK*. Please tell us how we can contact you directly so you don't miss a single issue.

Just a reminder: If you were a participating college last year, please be sure to update us with your current roster.

WE'RE HELPING AMERICA REGAIN THE COMPETITIVE EDGE. WE'RE HOPING YOU WILL TOO!

Rose Mansella, Circulation Coordinator
Circuit Cellar INK
4 Park St.
Vernon, CT 06066

Tel: (860) 875-2199
Fax: (860) 872-2204
Internet: rose.mansella@circellar.com
World Wide Web: http://www.circellar.com/

CONNECTIME

conducted by Ken Davidson

**The Circuit Cellar BBS
300/1200/2400/9600/14.4k bps
24 hours/7 days a week
(860) 871-1988—Four incoming lines
Internet E-mail: sysop@circellar.com**

I decided to stick with a single message thread again this month. In this discussion, we start by trying to solve a specific problem, but end up debating the merits of Sigma/Delta A/D converters.

Watch this space next month for some new special offerings for those of you who can't frequent the BBS as often as you'd like.

Micro- or Picoamp A/D

Msg#: 3309
From: Calvin Krusen To: All Users

I have a sensor which generates a current proportional to its stimulus. However, the output of the sensor is on the order of 100 pA to 10 nA. That's right, pico- and nanoamps! The cell of the sensor requires about 75 VDC, and has a signal that is quite noisy. The "noise" is not electrical, but part of the "cell's" response to stimuli.

I need to digitize the signal from the cell, after conditioning the signal, but I fear that the input impedance of gain-control op-amps will not be high enough and will degrade my signal.

Are there any op-amps specifically suited for this application? Also, how does a Sigma/Delta ADC differ from others?

Msg#: 3352
From: James Meyer To: Calvin Krusen

There is a configuration using an op-amp that is referred to as a "transimpedance" amplifier or a current-to-voltage converter. I believe you should explore that as your first stage connected to the sensor.

If your environmental specs-temperature and humidity, for example-aren't extreme, you could use something like a CA3 104, with the input current and voltage offsets adjusted with trimmer resistors to do the job. I'm assuming that you don't need microsecond response time along with subnanoamp sensitivity. You didn't say what frequencies you were working with.

Sigma-Delta A/D converters can be simple, cheap, and offer quite high effective bit resolution (20 bits is not uncommon), but they are not suited to measuring DC. The "delta" part is the giveaway. They require a constantly changing input voltage and average out any steady DC input to zero. They are quite effective converters for digital audio use, for example.

Msg#: 3360
From: Calvin Krusen To: James Meyer

What do you mean when you say they are not good for DC?

Will a steady DC signal provide a steady digital output, or will the output drift do to lack of "delta"?

Msg#: 3393
From: James Meyer To: Calvin Krusen

I *should* have said that I have never seen one applied to a DC source. I have since heard that you can use them for DC.

Msg#: 3372
From: Brad Sanders To: James Meyer

Jim, the "delta" function also comes from the input summing junction, which basically is an integrator feeding back on itself (i.e., a comparator constantly biased to the "noisy" part of the window).

Sigma/Delta chips can have DC response just like any other (as can Delta/Sigma D/A converters). Crystal Semiconductor, for example, makes a Sigma/Delta ADC designed specifically for high-accuracy, low-speed functions. It has resolution beyond 24 bits from DC to a few tens of hertz.

Msg#: 3396
From: James Meyer To: Brad Sanders

I jumped to an unwarranted conclusion based on the adaptive companding chips that Harris and Motorola make. Those are the only ones I have had hands-on experience with.

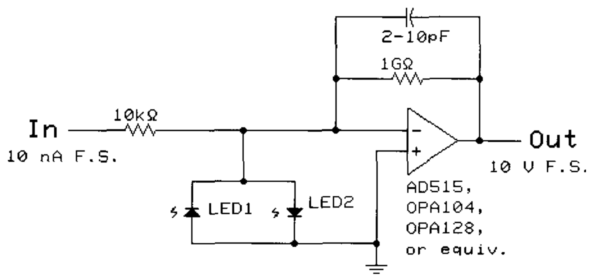
Msg#: 3378
From: Brad Sanders To: Calvin Krusen

That's something of an ill-conceived generalization spawned by the (ignorant) hi-fi press.

Fact is, there is no such requisite, and if there was, most CD players and DAT decks made nowadays would not have DC response (well, most don't, but it's just a coupling cap away.. .), nor would many EKG machines, seismographs, package scales.. .

Here's an extremely simplified "equivalent" of a Delta/Sigma front end:

CONNECTIME



Now, what do you get out!

"Noise." The comparator will swing rail to rail (let's say ±1 V), which is then subtracted from the input. This is then delayed in the integrator, which presents a sort of "rolling error" to the comparator. If this looks a little like a VFC, good, because it is. If one just "counts" the number of 1s then the number of 0s over a given time period, one can derive the input voltage:

Clock	E1	E2	"1 bit" output
0	0	0	0
	0.6	0.6	1
2	-0.4	0.2	1
3	-0.4	-0.2	-1
4	1.6	1.4	1
5	-0.4	1	1
6	-0.4	0.6	1
7	-0.4	0.2	1
8	-0.4	-0.2	-1
9	1.6	1.4	1
10	-0.4	1	1
11	-0.4	0.6	1
12	-0.4	0.2	1
13	-0.4	-0.2	-1

By averaging the values (0.2, -0.2, 1.4, 1.0, 0.6) we come up with 3.0/5, or 0.6 V. This will be true as long as the input remains at 0.6 V.

Delta/Sigma ADCs "work" by applying high-order IIR and FIR (mostly FIR) filters to this output. The higher the filter order and the higher the initial operating frequency of the input stage, the greater the resolution and frequency response one can derive from this "1-bit" output.

These ADCs exploit the fact that a given level of noise (quantization) becomes less and less significant over a wider bandwidth. A Nyquist-sampled "1-bit" ADC will have nearly a whole 8-dB SNR, but each time the sample rate is doubled, you can add another 6 dB to this ratio (same noise energy, double the bandwidth), so:

$2f_s$	$4f_s$	$8f_s$	$16f_s$	$32f_s$	$64f_s$	$128f_s$	
13 dB	19 dB	25 dB	31 dB	37 dB	44dB	50 dB	etc....

Now, this is all ***very*** simplified. Practical ADCs use noise shaping to further increase "audible" SNR, and may have several cascaded stages (like the above) which further increase resolution, but this is basically how a Delta/Sigma ADC works.

It should also be clear that if the input voltage does not change (i.e., DC] the output "duty cycle" will not change.

Msg#: 3337
From: George Novacek To: Calvin Krusen

The simplest approach is identical to the ionization smoke detector, where the ionization chamber current is in the order of 10 pA.

Using a load, you convert the current to a voltage, which is than processed by a high-input-impedance op-amp; you need a MOSFET without a diode-protected input.

For the load, smoke detectors usually take advantage of two chambers connected in series. One detects, and the other serves as a constant load (which, however, can compensate for environmental changes).

As an alternative, there are high-value resistors available to use instead. You will need something on the order of 10 GΩ.

For the amp, the simplest approach is to buy a smoke detector IC (Motorola), provided you keep the input voltage swing within allowed limits. With potentially a 75-V swing, you may have to consider a MOSFET in a source-follower configuration, which will have a voltage gain a bit less than 1, but will give you output voltage at a kilohm impedance level.

Depending on the swing you need to work with, signal bandwidth, and so forth, you should be able to select a load which will work quite well. I would still use an op-amp between the input MOSFET and the digitizer as a buffer.

Msg#: 3493
From: Pellervo Kaskinen To: Calvin Krusen

Yours is a classic case of an application for the transimpedance amplifier, with the possible exception of a need for input protection. The 75-V supply to your sensor causes this requirement.

National Semiconductor, Burr-Brown, and others sell op-amps with under 1-pA bias currents. In fact, I recall Burr-Brown last year running several ads about an amplifier that has only 15-fA bias current specification.

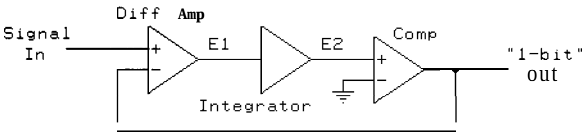
The more difficult part is finding good feedback resistors and building the circuit. Bob Pease of National has covered several of the traps you might fall into in his column "Pease

Porridge” in ***Electronic Design.*** Too bad, I did not find a reprint of this particular topic in either his or Jim Williams’ book.

Anyway, if your maximum signal current is 10 nA and you’d like to have a 10-V full scale, we are talking about a 1-GΩ resistor. You can get them from Dale, Victoreen, or Caddock.

Now, you are worried about the noise. But all you need is just one picofarad over the feedback resistor and you have a low-pass filter of 160 Hz! Careful, the leakage in that capacitor might reduce your resistance..

The protection I mentioned probably requires an input resistor that you would not need in a true transimpedance amplifier. The overall diagram is something like:



Note that the output is negative if the input current is positive.

The protective diodes are actually GaAs-type LEDs that have a lower leakage than any silicon diodes. You could also use FDH300 low-leakage diodes from National (used to be a Fairchild product), but a decent LED is easier to find.

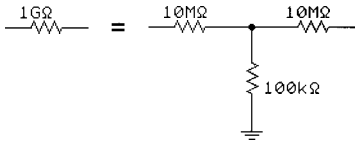
I have increased the feedback/filter capacitor value because the input protection diodes have capacitance that would cause bad ringing without adequate filtering capacitor values. I believe 2 pF to be enough, but you may want a lower filtering frequency anyway.

As far as the Sigma/Delta ADC goes, it is just one A/D configuration that is beneficial in some applications, mostly in the audio. Rather than producing one multibit conversion against an “absolute” reference, it continuously converts the difference (delta) between the input and the accumulated output (sigma). It does this with only a single bit of resolution at a time.

An input step function would produce a whole lot of 1s until the sigma finally reaches the input and produces the first 0. A sine wave is easier. The conversion continuously follows the slopes. That explains part of the Sigma-Delta converter’s appeal in the audio field.

Msg#: 3554
From: James Meyer To: Pellervo Kaskinen

Replacing the single 1-GΩ resistor with a “pi” network reduces the resistance values to manageable proportions.



Shunt capacitance is also made more manageable.

Msg#: 3660
From: Pellervo Kaskinen To: James Meyer

For a while, I contemplated including the 3-resistor network, but then chose not to. The main reason is the bad effect it has on the offset error. Analog Devices presents the pros and cons in their AD5 15 data sheet or some app note, if I recall.

Msg#: 3496
From: Brad Sanders To: Pellervo Kaskinen

You know, the only real “advantage” Sigma/Delta ADCs and DACs have in audio is price. Because these are *far* cheaper to produce in quantity (not just the chips, but the entire circuit), one can get nearly 20-bit performance from very inexpensive (OK, cheap) equipment.

PK> .An input step function would produce a whole
PK> lot of 1s until the sigma finally reaches the input and
PK> produces the first 0. A sine wave is easier, the
PK> conversion continuously follows the slopes. That
PK> explains part of the Sigma/Delta converter’s appeal in
PK> the audio field.

Pellervo, I’m surprised! A step function into a Sigma/Delta ADC isn’t going to produce any more 1s and 0s than a DC function.

Indeed, any Sigma/Delta front end is going to be used with a corresponding digital filter (most likely on the same “chip”), so it becomes quite irrelevant (except from a purely audiophile, “spiritual” POV) which conversion method is employed.

If you want very high-speed conversions, you use R2R ladders or flash ADCs. If you can run slower, you use Sigma/Delta, or even a combination of technologies (like dual slope or “oversampling” via the microcontroller’s built in low-resolution ADC).

This whole “AC thing” is quite nonsensical. A sine is no “easier” for a Sigma/Delta converter to handle than is DC, or any other (bandwidth limited) waveform.

This all started when Julian Hirsch (or one of his highbrow audiophile doppelgangers) tried (badly) to explain that

"DC" from an R2R **DAC** is limited by its (fairly poor) thermal properties, whereas "DC" [or any other waveform) from a Delta/Sigma DAC is limited by the accuracy of its internal digital filter and the external low-pass filter [because a Delta/Sigma DAC doesn't "do DC". Instead, it does noise-shaped PWM AC).

While I'm not accusing you of listening to Julian Hirsch, this topic is somewhat of a pet peeve of mine, and I'd hate to think someone actually picked up this nonsense on a visit to the one of the last bastions of "truth, liberty, and the gearhead way"-the Circuit Cellar.

Whether I have a CS5390 or an AD1876, a 1-VDC input is going to get me a string of 1s and 0s that corresponds to -1 VDC and repeats at the sample rate. The only difference is the CS5390 is going to give me nearly 20 bits accuracy (vs. nearly **16** bits) with far less "dither" from (internal) thermal noise than will the AD1876.

Msg#: 3658
From: Pellervo Kaskinen To: Brad Sanders

I'm sorry to have caused such heartburn and disappointment!

Yeah, I was a bit too quick in my choice of words. What I should have elaborated a little more is the internal nature of the **"1"** stream. The real output, of course, is oblivious to them. But the fact remains, the 1 -bit converter (comparator) produces fewer consecutive 1s after settling to DC than it does on a step.

These details notwithstanding, the audio-type Sigma/Delta converters are specified for linearity, not absolute full-scale value or even zero offset stability. While they may be every bit as good as the "industrial" ADC/DAC components, the specification remains the difference, at least today.

Msg#: 3683
From: Brad Sanders To: Pellervo Kaskinen

OK, Pellervo, let's just be clear on this: "audio type" and "Sigma/Delta" are not universal associations. "Audio type" components of *either* flavor value different specs than do other data-acquisition devices. As always, it's just a matter of choosing the right part for the job.

Consider that one of the first cost-effective 24-bit (instrumentation) ADCs was the Crystal Semiconductor part (a company that makes its bread and butter in audio devices). This part-with very tight specs on absolute accuracy, full-scale error, offset stability, etc.-is a Sigma/Delta device.

Lately, Analog Devices has come out with its own new "high-accuracy" (24-bit) parts intended for precision applications-which also just happen to be Sigma/Delta converters.

An R2R (or charge transfer) ADC is going to be limited by things like divider linearity, thermal drift, comparator accuracy, and so forth.

A Sigma/Delta part is going to be limited by the accuracy of a single-bit DAC, the comparator used in the front end, and internal (digital) filter precision.

Both have their tradeoffs, but neither is inherently more accurate.. .not even for "DC."

Msg#: 3874
From: Pellervo Kaskinen To: Brad Sanders

I'm familiar with the Crystal offering. I also know of at least one other similar product (i.e., non-audio-specific Sigma/Delta). But that's it. All the myriad other offerings are specified by linearity and noise, not by DC parameters.

We *invite you to 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 (860) 871-1988. Set your modem for 8 data bits, 1 stop bit, no parity, and 300, 1200, 2400, 9600, or 14.4k bps.*

ARTICLE SOFTWARE

Software for the articles in this and past issues of *Circuit Cellar INK* may be downloaded from the Circuit Cellar BBS free of charge. It is also available on the Internet at **http://www.circellar.com/**. For those with just E-mail access, send a message to **info@circellar.com** to find out how to request files through E-mail.

For those unable to download files, the software is also available on disk. Software for issues prior to 1995 comes on a 360-KB IBM PC-format disk, one issue per disk. For issues from 1995 on, software comes on a 1.44-MB PC-format disk, with three issues per disk. Disks cost just **$12** each. To order Software on Disk, send check or money order to: Circuit Cellar INK, Software On Disk, P.O. Box 772, Vernon, CT 06066, or use your Visa or Mastercard and call (860) 8752199. Be sure to specify the issue numbers with your order. Please add $3 for shipping outside the U.S.

431 Very Useful **432 Moderately Useful** **433 Not Useful**

PRIORITY INTERRUPT

Software-the Real Generation Gap

I think we hardware guys have been mute too long. Have you looked at the state of affairs lately and realized how one-sided the "computer revolution" has really been?

I never criticize a person for unnecessary equipment upgrades. After all, for every computer-equipment fanatic, you'll find an equally obsessive car owner, like me. But, are people who upgrade from a '486 to a Pentium to a Pentium Pro in 2% years helping the economy or perpetuating a constant evolution of inefficient programming and vaporware from developers?

In the 20 years I've been in the computer business, I've counted eight generations of significant improvement in CPU design and hardware architecture. The hardware has become so reliable, in fact. that we take it for granted. When it comes to software, on the other hand, the word "generation" has come to only mean the latest release. independent of whether there is any actual improvement. In my opinion, most of what we realize as performance enhancements has come from increased CPU horsepower and reduced memory-access time.

PC software has become bloated and slow. Tell me that the word processor you had on your '286 wasn't more clear-cut and unambiguous than the one you use now. With the latest software releases. you need at least a Pentium so you don't go to sleep while it saves to disk or searches a file. Worst yet, while the older version couldn't anticipate your thoughts or use its modem link to the Library of Congress to write what you would have written had you actually pressed the keys, it at least had clear and predictable responses to executed commands. Today, it's not unusual to press a key and find your text mirror imaging itself, rotating counter-clockwise through a rainbow of changing fonts, or zipped down so many times that the Declaration of Independence becomes a single 80-character line. No wonder it has to have an Undo command.

Since *INK's* focus is embedded control, I shouldn't really care what happens to PC software, should I? Ordinarily not, but I'm seeing a disturbing trend.

Unlike the PC-processor craze, embedded-control designs are generally application, rather than performance-hype, driven. Regardless of the six-orders-of-magnitude processing advantage of a '486 over a lowly 8-bit processor, only the latter makes sense as a soda-machine coin counter. However, coordinating the coin counts from 500 soda machines is a job for a '486. It's the ability to select embedded controls from across eight generations of hardware that keeps controllers cost-effective and in perspective.

The bad news is that PC-based development systems, and especially embedded PCs, are going the other way. The whole idea behind the embedded PC was that the development and target systems shared similar hardware and software environments. You could develop code on your PC under DOS and then execute it using a DOS kernel on the target. A pretty straightforward idea and a very successful market.

Now the idea is to eliminate DOS and run only under a windowed environment. That move suggests that you either have a target system with memory, processing, operating system, and display attributes similar to the development system or keep old development software running because it's the only thing still generating code for a simple V25 board with an LED display.

For me, DOS or straight assembly language still seems a more logical choice for a simple V25-based solenoid actuator. If accomplishing this means leaving that old '386 and its 5-year-old development software unimproved but ready to use, so be it.

Steve

steve.ciarcia@circellar.com