EMBEDDED PC SPECIAL SECTION: EMBEDDED PC

CIRCUIT CELLAR INK®

THE COMPUTER APPLICATIONS JOURNAL

#80 MARCH 1997

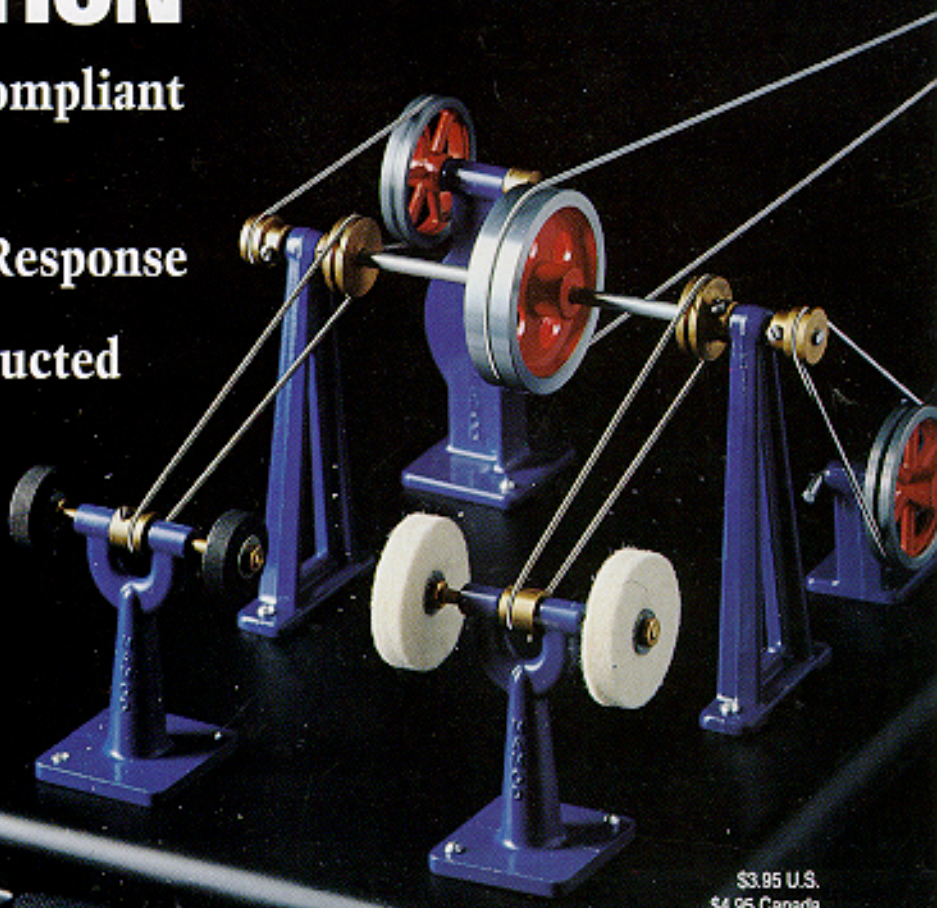
HOME AND BUILDING AUTOMATION

Building CEBus-Compliant Products

RF Voice-Control Response

Radiated and Conducted EMI Testing

EPC: LAN Media and Protocol

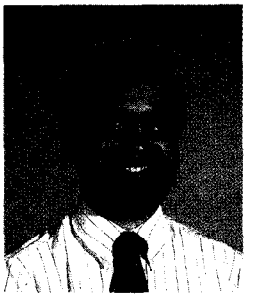


03
0 74470 75349 0

$3.95 U.S.
$4.95 Canada

TASK MANAGER

Busy Bees



I'm going to skip my usual insightful proselytizing this month because of how much we have going on these days. I haven't seen the Cellar hopping like this in years.

ConnecTime

I'll start with something close to my heart-the passing of ConnecTime. If you haven't been back there yet, be sure to read my introduction to this month's column for an explanation of what's going on. In a nutshell, we're adding full-bore Internet access to the system, so we felt ConnecTime's useful days were numbered. I decided to end it on a high note rather than let it peter out. Now that more of you will be able to join us on-line, be sure to check out what's happening on the BBS.

Design Contest

After eight years, the Circuit Cellar Design Contest has gone big time with some big money prizes. For the first time, the contest focuses on 'x86 architecture. Check out the details on page 55 of this issue or on our Web site. I want to thank Janice for all her work in pulling together the new-and-improved Design Contest.

The Issue

This year's Home and Building Automation issue starts off with the latest in what's going on with CEBus. I've followed the standard for years now, and perhaps with the recent changes that Adrian Dunn discusses, we'll soon see some affordable products.

Until CEBus products are available, we'll all still make heavy use of X-10 devices for power-line control. In the next feature article, Chris Arndt describes how to add remote X-10 control using a hand-held amateur radio unit.

For our last feature, John Morley shows how to add automation to an in-ground sprinkler system using the HCS II and a bit of custom engineering.

This month's columns begin with Part 2 of Joe DiBartolomeo's series on designing for EMI. In this installment, he discusses design and testing issues related to radiated and conducted emissions. Next, Jeff brings the old floating needle into the twentieth century with a quick interface to an off-the-shelf electronic compass module. Finally, Tom checks out the latest in content-addressable memory.

As for Embedded PC, Mike Baylis begins with an overview of how the trusty old STD bus has grown up into a 32-bit powerhouse. Next is Part 2 of Bill Payne's networking primer series. Bill looks at some of the protocols used on LANs. In PC/104 Quarter, Jonathan Miller compares and contrasts some of the many analog I/O boards available for PC/104 stacks. And lastly, Fred sets up his own embedded BBS that favors data acquisition over private messages and file downloads.

editor@circellar.com

CIRCUIT CELLAR®

THE COMPUTER APPLICATIONS JOURNAL

EDITORIAL DIRECTOR/PUBLISHER
Steve Ciarcia

EDITOR-IN-CHIEF
Ken Davidson

MANAGING EDITOR
Janice Hughes

TECHNICAL EDITOR
Elizabeth Laurençot

ENGINEERING STAFF
Jeff Bachiochi

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

CONTRIBUTOR:
Pelletvo Kaskinen

ASSOCIATE PUBLISHER
Sue Hodge

CIRCULATION MANAGER
Rose Mansella

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 (660) 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 531.95, all other countries $49.95. All subscription orders payable in U.S. funds only, via international postal money order or check drawn an 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 0 Box 698, Holmes, PA 19043.9613.

Circuit Cellar BBS-24 Hrs. 300/1200/2400/9600/14.4k bps, 8bits, no panty, 1 stop bit,(860)871-1988. For information, mail to **info@circellar.com**. World Wide Web: **http://www.circellar.com/**

EMBEDDED PC

See Pages 35-64 for our Special Section

INSIDE ISSUE 80

READER I/O

THE RIGHT DIRECTION

I recently enjoyed reading Do-While Jones' "The Global Positioning System" (INK 77 and 78). His explanations are clearer and more understandable than any other references I've come across.

Mark Pham
pham_mark@lmsc.lockheed.com

A LITTLE TOO MUCH HOPE

Steve's "Hardly a High-Definition Opinion," (INK 78) shows too much optimism. Microsoft could not declare a standard. It isn't even consistent with itself.

Try hardcoding in the DOS kernel. All sorts of stuff is supposed to come from fields in the partition table and initial disk control block, but it's ignored.

I firmly believe that Microsoft releases as follows: Alpha-Aw, it sort of works, Beta-But it's supposed to work, and Gamma-Gotcha! We want loot for the fixes.

Stanley Taylor
sftaylr@cts.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) 871-0411.
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.** corn. 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

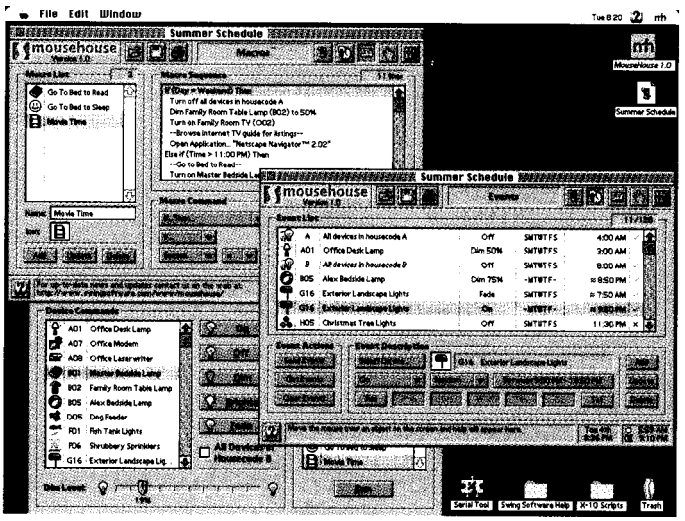
Edited by Harv Weiner

X-I 0 SOFTWARE FOR MACINTOSH

MouseHouse allows the control of electrical appliances in a home or office using the industry standard X-10 protocol. The software enables appliances to be turned on and off, lights dimmed, weekly events scheduled, and command sets constructed via a single mouse click.

MouseHouse sends commands through a low-cost CP290 controller plugged into a serial port on a Macintosh computer. The commands are intercepted by X- 10 modules placed between appliances and the wall outlet. Software features support multiple house files for creating season-specific schedules, random scheduling of events, and sunrise / sunset calculations. Programmed events continue to work when the Macintosh is off. Context-sensitive online help is available, and it operates on any Macintosh (accelerated for Power Macs). MouseHouse is compatible with the Macintosh Communications ToolBox, so it can support electronic serial port switches and the Macintosh Geoport.

MouseHouse is available for $69.95 from various home-automation resellers or for $59.95 from Swing Software. It requires a **CP290** Home Control Interface and at least one X-10 module. The CP290 can be purchased for about $35, and X-10 modules are found for as little as $10 at Radio Shack or a home-automation retailer. A free demo is available on Swing Software's Web site.



Swing Software
P.O. Box 147
Romeo, MI 48065
(810) 336-9000
Fax: (810) 336-0116
http://www.swingsoftware.com/mousehouse/

#500

SPEECHDEVELOPMENTSYSTEM

Quadravox has announced a PC Speech Development System called Qbox. The system consists of a Windows sound-processing software package and a chip programmer that connects to a PC parallel port.

In a three-step process, you can add your own phrases, sentences, and sounds to any electronic project. First, you record and compress your own .WAV files, edit as necessary, and define desired sound sequences. The total recording time is 1 min. for up to 128 sounds or phrases and up to 128 user-defined sequences. An exact, real-time software simulation of the synthesizer ('486/66 or better PC required) lets you hear the output speech before chip programming.

The data for the compressed sounds and user-defined sequences are programmed into a Texas Instruments TSP50P11 chip through the PC parallel port. The programmed chip is plugged into an application module, which has keypad, serial, and parallel interfaces. A home-automation X-10 interface is also available.

The Qbox PC Speech Development System hardware and software sells for $90. Preassembled application modules are available for $25. Complete hardware and software documentation is supplied in the form of Windows help files with schematics. Prerecorded vocabulary is available at no extra charge. Free demo software can be downloaded from the vendor Web site.

Quadravox, Inc.
1701 N. Greenville Ave., Ste. 608
Richardson, TX 75081
(972) 669-4002
Fax: (972) 437-6382
http://www.quadravox.com/

#501

NEW PRODUCT NEWS

TELEPHONE INTERFACE MODULE

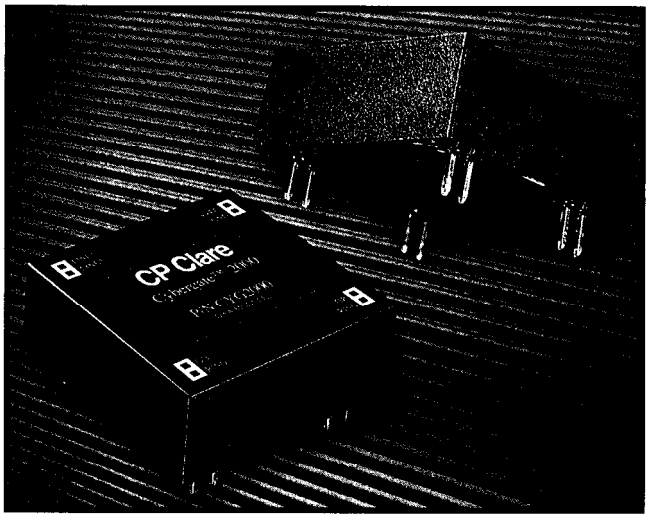
A fully integrated modular Data Access Arrangement (DAA) is available from CP Clare. The **Cybergate CYG2000/CYG2001** combines ring detection, hookswitch, and gyrator/transformer functions, as well as surge and transient protection in a 1 .07″ x 1.07" x 0.4" plastic module that provides the necessary galvanic isolation. The unit is FCC Part 68 compatible, eliminating the need to design a specific FCC-compliant DAA.

Using transformer coupling for the signal path and optocouplers for the ring detection and on/off hookswitch circuits, the unit offers 0.01% THD, 1500-V peak isolation, and 300-V surge protection. It accepts ring voltages as high as 150 Vrms. Optically coupled MOSFETs implement a 7-mW hookswitch that features a maximum On resistance of 15 Ω and a maximum switching current of 120 mA. The CYG2000/CYG2001 employs an innovative electronic gyrator circuit to minimize the DC resistance. This same circuit also maximizes the AC impedance seen by the telephone line when the unit's hookswitch is closed.

The DAA can be used in telephone interfaces for home, medical, and industrial processes, as well as for security, fax, voice-mail, computer, and modem applications. The CYG2000 (half-wave ring detection) and CYG2001 (full-wave ring detection] each sell for $12.57 in 5k quantities.

CP Clare Corp.
Semiconductor Gr.
78 Cherry Hill Dr.
Beverly, MA 01915
(508) 524-6700
Fax: (508) 524-4910
http://www.cpclare.com/

#502



PC CONTROLLER

The MegaSwitch Model VIP-1322 switch can control two PCs using one keyboard, mouse, and monitor. It's ideal for situations having limited workspace, such as data acquisition, industrial control, multiple-platform graphic applications, and software development.

MegaSwitch's features include selection of the active PC by keyboard key sequence, front-panel indicators showing the selected PC, separate indicators showing the PCs' power status, memory for keyboard LED status, error-free bootup, and mouse-driver loading. An electronic interlock prevents selection of a PC that isn't powered on.

Intelligence in the MegaSwitch ensures that operation through the switch is completely transparent. It handles all required keyboard and mouse responses during bootup of the connected PCs and during OS load. No keyboard errors are generated, and mouse drivers load correctly. Switching of the keyboard and mouse is entirely electronic and does not disturb or affect programs running in the different PCs. Either PC can be independently powered up or down at any time.

The keyboard, monitor, and mouse connect to the input of the VIP-1322 with their normal cables. Standard extension cables connect the MegaSwitch to the keyboard, mouse, and video ports of the two PCs.

The Model VIP- 1322 measures 8" x 6.25" x 2.5" and sells for $360 in single quantities.

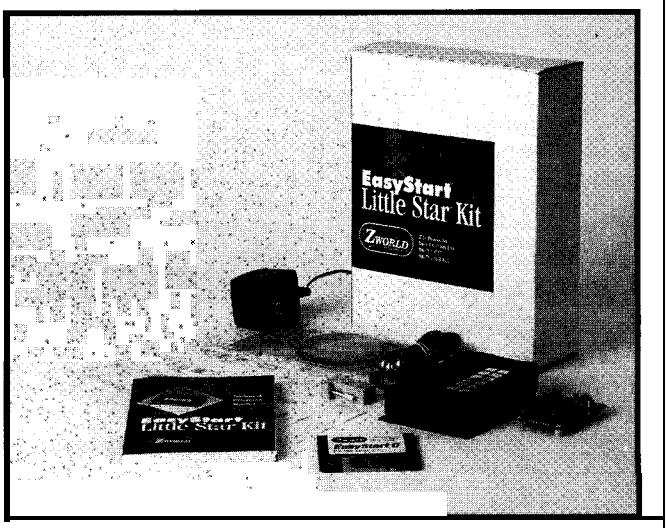
Vetra Systems Corp.
275-J Marcus Blvd.
Hauppauge, NY 11787
(516) 434-3185
Fax: (516) 434-3516
http://eemonline.com/vetrasyst/

#503

NEW PRODUCT NEWS

SOFTWARE DEVELOPMENT SYSTEM AND HARDWARE KIT

Z-World's **EasyStart** Kit combines the EasyStart C software development system with the Little Star controller to provide a fast and easy way to program multitasking control systems. The Windows-based EasyStart C (a simplified version of C) is an integrated software development system that includes an editor, compiler, and debugger. With EasyStart C, new users can write control programs immediately, without mastering all the intricacies of standard C. The accompanying manual and software library contain numerous practical examples and a self-teaching tutorial.



The Little Star controller has 16 TTL-compatible digital inputs and 14 high-current digital outputs suitable for driving small relays and other actuators. The controller is housed in a rugged enclosure and features a 9.216-MHz Z180 processor, 128 KB of nonvolatile flash memory, and 32 KB of battery-backed SRAM. A version is also available with a 2 x 12 keypad and a 2-line x 20-character LCD. With the supplied software, you can easily add network controllers, controllers, and expansion boards to increase the number and type of I/O channels.

The Little Star EasyStart Kit includes a choice of controller (simple enclosure or LCD and keypad), manual, software diskette, demonstration board, power supply, and cabling. Unlimited technical assistance is free.

The Little Star EasyStart Kit (standard enclosure) sells for $389. The Little Star ESX Kit (LCD and keypad) sells for $489.

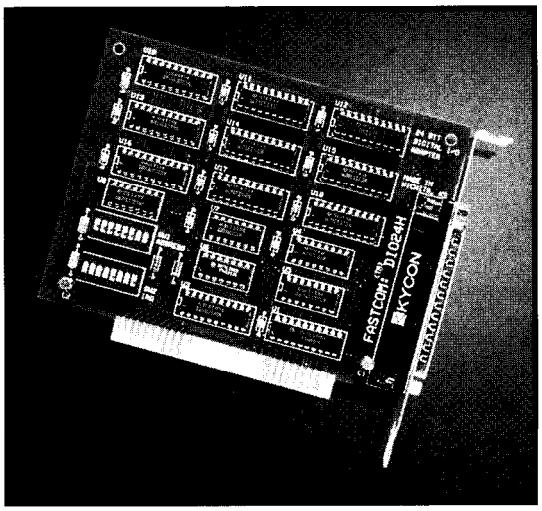
Z-World
1725 Picasso Ave. • Davis, CA 95616 • (916) 757-3737 • Fax: (916) 753-5141 • **http://www.zworld.com/** **#504**

24-BIT DIGITAL I/O CARD

The Model **DIO24H** is a low-cost, general-purpose, 24-bit DIO adapter board for IBM PC- and AT/386-compatible computers. The board provides a high-current sink output of 64 mA for direct excitation of up to twenty-four 5-V relays. Additionally, its shared interrupt capability enables users to share a single IRQ line with multiple boards in the same PC.

Typical applications include process monitoring and control, building security systems, and other systems requiring relay contact closures in which processing speed is of secondary concern to the capability of directly interfacing with relay outputs.

The DIO24H provides three 8-bit bidirectional ports and two input lines for interrupt Enable and Interrupt. The interrupt Enable line may be used as a failsafe to ensure that an interrupt signal is only generated as an intentional event. One of the three ports may be partitioned into dual 4-bit ports-one high and one low. Port direction and configuration is specified by an onboard control register.



Operational specifications include a range of input low and high voltages of -0.5 to +0.8 VDC and 2-5 VDC, respectively. Output low is +0.55 VDC maximum and, on the high end, is at least 2.0 VDC at 15-mA source current.

The DIO24H sells for $125.

Industrial Computer Source
9950 Barnes Cyn. Rd.
San Diego, CA 92121
(619) **677-0877**
Fax: (619) 677-0615
http://www.industry.net/indcompsrc/

#505

FEATURE ARTICLE

Adrian Dunn

Building a CEBus-Compliant Product

With the advent of the CEBus as a full-fledged EIA standard and of chips from at least two sources, CEBus can now rise to the whole-house communications protocol it was intended to be. Pop in for the update.

Technology enabling the reality of home automation has been long awaited by both developers and manufacturers. How many times have we heard the promise that such a future is just over the horizon?

Although devices that automate your sprinkler system or connect a closed-circuit camera to your TV crop up with alarming regularity, a reliable, affordable, extensible, and user-friendly whole-house communications protocol never seems to surface.

Many hoped the CEBus Standard would fill this gap but were disappointed when few products appeared since the standard's inception five years ago.

However, this lack of products makes sense. Until recently, CEBus was an EIA interim standard. Changes could be introduced that might make previous products incompatible with newer ones.

As well, only one source of chips implemented the standard. And, they were simply too expensive for the margin-conscious home market.

Now that the CEBus Standard is in the final stages of ANSI balloting to become a full-fledged EIA standard and sub-$10 chips will be available from at least two sources in 1997 [i.e., Intellon

and Domosys), it appears that a whole-house communications protocol has arrived. Developers and manufacturers are beginning to seriously consider the enormous potential of the home-automation market.

WHAT'S THE CEBus STANDARD?

It's an open communications protocol specifically designed for networks linking residential consumer and utility products. It specifies how different products communicate with each other, what they say, and over which media.

Although seven media are specified, development efforts concentrated on designing chips and products for the power-line (PL) and radio-frequency (RF) media. Neither require additional wiring to be installed in homes.

The ultimate goal of the CEBus Standard is to enable consumer products to be added incrementally to a residential communication network with little or no external support. As a reliable, low-cost, and manufacturer-independent protocol, it ensures product compatibility.

The organization chosen by the EIA to promote the standard-the CEBus Industry Council (CIC)—is currently setting up a center to test and certify conformity to the CEBus Standard. Certified products will be labeled as CEBus compliant.

From a protocol standpoint, the standard implements four of the seven OSI layers-the Physical, Data Link, Network, and Application layers. Several message-delivery services are specified at the Data Link and Application layers. Their use depends on the application.

More reliable services acknowledge whether a message is successfully delivered or not. Therefore, you can retransmit the message if necessary, ensuring 100% reliability for critical devices (e.g., security systems).

Control channels transmitting 10 kbps are specified for every medium. The coaxial and the twisted-pair media also have defined data channels, which are described in terms of a certain bandwidth allocation.

No data formats are specified for these channels, so manufacturers can conveniently allocate bandwidth. Telephone conversations, music, and compressed video can be transmitted across the same twisted pair via its own data format and allocated bandwidth.

Authentication services defined at the Application layer preserve sensitive information (e.g., passwords or utility rates). A message-segmentation service defined at the Network layer lets you transmit long messages. And, the protocol still has room for future services not thought of yet.

The list of services is extensive. With input from over 300 companies, the CEBus Standard paves the way for a broader market and value-added products.

WHAT IS CEBox?

As you see in Figure 1, CEBox is a software integration tool that implements the CEBus Standard. It enables developers and manufacturers with little or no knowledge of the protocol to develop a CEBus-compliant product rapidly and cost effectively.

The CEBox application medium is the physical means for CEBus-compliant communication. The standard specifies five communication media besides PL and RF-twisted pair (TP), coaxial (CX), fiber optic (FO), infrared (IR), and Audio/Video Bus (A/V Bus).

The hardware component of any CEBus-compliant application-the transceiver-transmits and receives messages over the chosen medium. It can fully implement the lower two layers of the CEBus application model (Physical and Data Link). The CEBox development platform can be ported to any CEBus-compliant hardware.

CELib COMMLINK INTERFACE

The CELib Commlink Interface (CCI) is a software module that abstracts communications differences between various transceivers. CEBox contains a different CCI module for each supported transceiver.

CELib is a software implementation of the CEBus Application and Network layers as well as the Layer System Management (LSM). Although it's the heart of CEBox, it's modularized. Your compiled application only includes the portions necessary for your project.

USER APPLICATION

On top of these segments and components lies the User Application and the CEBox-generated code representing the implemented Contexts, Objects, and Instance Variables (IVs). The 'necessary sections of code can be divided into three categories:

- the `main()` routine—where program execution begins
- the I/O routine interface—it lies between the hardware and the IVs.
- the USR routines-access nonvolatile memory and provide feedback to the User Application. This section is called by CELib.

CEBox furnishes a template for each routine category which you can flesh out with your own code. From the User Application, *you* can call the library of CELib interface functions.

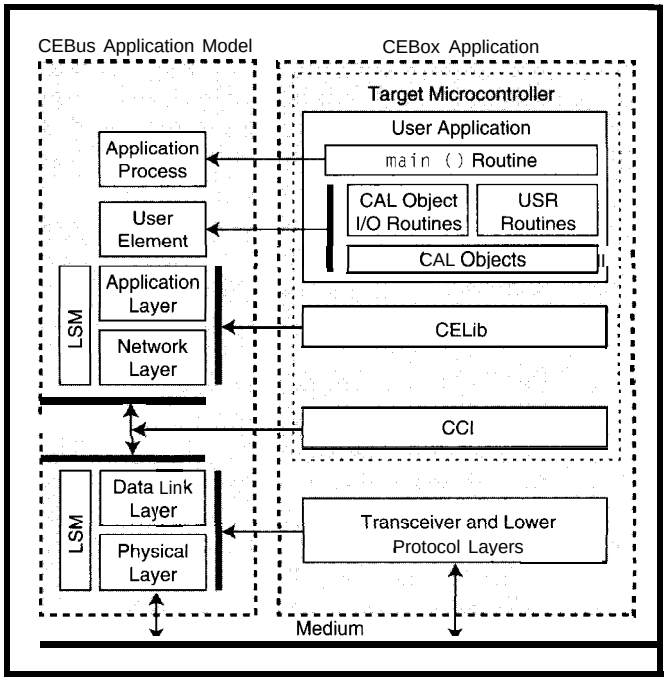


Figure 1—*The* program on the *target microcontroller can be divided info three sections-the manufacturer's application, the CEBus Application and Network layers (CELib), and a module that interfaces with the selected CEBus transceiver (CCI).*

These functions can be subdivided into four functional categories-address acquisition, IV access, CEBus communication, and miscellaneous. Note that the User Application, CELib, and CCI all run on the same target microcontroller.

CELab PROTOTYPING

CEBox comes with CELab, a hardware tool with three sections, as you see in Figure 2. It contains circuits for communicating with a PC as well as the program and data memories used by the target microcontroller.

CELab is where the user performs all prototyping tasks. It has a default 80.51-type target microcontroller (e.g., an 80CE558) and several peripherals, including an LCD, a battery-backed clock, four LEDs, and six switches.

You can remove this board from CELab without turning off its power. So, you can modify a circuit in the prototyping area without reloading software into the main-board program memory.

Blank prototyping boards are also available. Future versions of this board will be made for other popular microcontrollers.

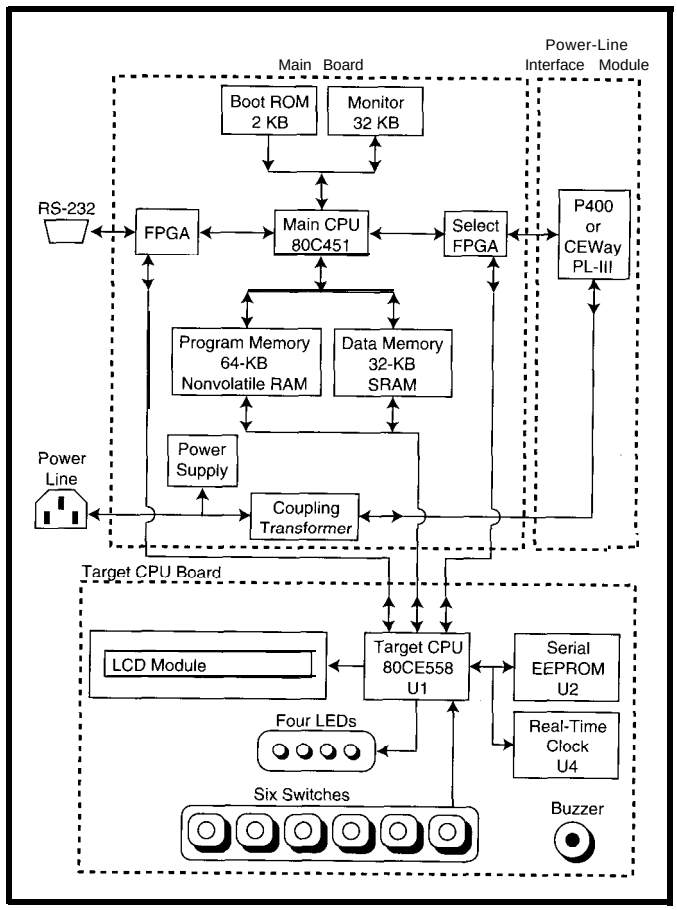


Figure *P-The CELab hardware prototyping environment comprises three modules. The main board handles PC communications, and the CEBus transceiver is on the power-line interface module,* while *the target CPU board contains the target microcontroller and hardware peripherals.*

POWER-LINE INTERFACE MODULE

This module contains a CEBus-compliant power-line transceiver such as the Intellon CENode-CEThinx and P400 modules or Domosys CEWay PL-III modules (to be released this year).

Now, with all the parts in place, I'll demonstrate how to use the CEBox tool to develop a clock that broadcasts the current time throughout the house.

BUILDING A MASTER CLOCK

Who hasn't confronted the blinking 12:00 of an unset clock, be it on your VCR or stereo? Have you missed an appointment because a power outage unset your alarm clock!

Who hasn't wished for a no-brainer solution so each time the power went out or you had to move your VCR, the clock would come back on with the right time?

My goal is to show you how to implement a master clock that uses CEBox to regularly broadcast the time over power lines to all CEBus-compliant devices in the house.

Although I show it as a stand-alone product, it's better to incorporate it as part of a VCR or television. It's potentially valuable to differentiate one TV or VCR from the rest of the field.

CHOOSING YOUR CONTEXTS

The CEBus Standard provides a versatile and modular structure for representing hardware. At the top level, Contexts represent devices or device subsystems (e.g., Time, Light, and Coffee Maker-honest!).

Each Context is composed of one or more Objects (i.e., particular functions or controls within a Context). Each Object comprises one or more IVs, representing items within an Object.

Photo 1 shows CEBox's Context Builder, a tool to select which CEBus-standard Contexts, Object, and IVs to implement within a device. All devices require the Universal Context. Within its Node Control Object, IVs hold information pertaining to the whole device.

The only other Context I needed was the Time Context. It contains several time-oriented Objects (e.g., Alarm and Time Schedule).

The only nonmandatory Object I implemented was the Real-Time Object. It contains IVs storing the current time.

Other nodes can interrogate the master clock for the current minute or month, and these are returned in a numeric format. Even better, they can have all the necessary information returned via a string IV called `current-time`.

This string contains the year, month, day, hour, minute, and second in ASCII format. Day of the week is represented as a bit string.

However, the most useful IVs in the Real Time Object are the four reporting IVs. Many Objects contain them, and if implemented, they generate a CEBus-compliant message when a particular condition evaluates True.

The `reporting_condition` IV specifies the boolean condition to be evaluated. I wanted the `current_time` IV value to be broadcast every 5 min., so it is expressed as:

'C' AND 'm' DELTA '5'

This boolean expression has two separate relational expressions—{'C'} and ('m' DELTA '5') (see Table 1).

{'C'} consists of the label of the `current_time` IV and evaluates True as long as any ASCII string is present within `current_time`. This expression serves a syntactic purpose. The first argumentin `reporting_condition` must be the IV to be reported. Usually,

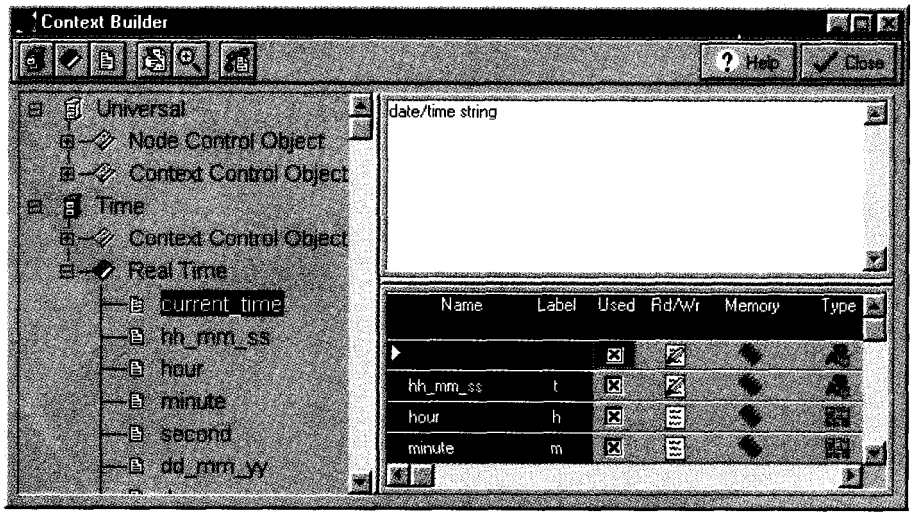


Photo I-Use *CEBox's Context Builder to select which CEBus Contexts, Objects, and IVs you wish to implement in your product. The IV Grid in the bottom right-hand corner of the Context Builder lets you change the attributes of the implemented IVs.*

this IV is the one being tested, so there's no need for a second relational expression.

However, to report the value of the current_time IV, I had to test the minute IV. The second relational expression tests to see whether minute has changed by 5 since the last time the DELTA expression evaluated True.

The previous_time IV initializes itself to the current value of the IV used in conjunction with the DELTA operator in report_condition (i.e., 'm' or minute). Thereafter, the value only changes whenever the DELTA operator generates a report. At that point, previous_value updates with the current value of the IV under test.

report-address is a four-byte IV specifying where to send the report. The first two bytes are the destination unit-address, while the second two are the destination system_address.

This IV was initialized with broadcast addresses (0x0000) for the destination system and unit addresses, so all nodes on the same network received my reports. Once the clock is plugged in and gets its own system_address, the code modifies this IV so reports are only sent to nodes with the same system-address.

The report-header IV specifies which IV in the destination node(s) should receive the report (see Table 2). Although, in this case, it's the same IV reported (i.e., the current_time IV of the Real Time Object), this isn't typically true for other reporting objects.

These four IVs are stored in RAM. So, another node can easily change the frequency or the destination of the reports simply by sending a SetValue message to change the value of one of these IVs.

COMMUNICATION PARAMETERS

The CEBus Standard offers many possible transmission services that can be somewhat bewildering at first. CEBox simplifies the selection of the default parameters because its windows help you configure the Data Link, Network, and Application layers of your node.

Many parameters don't need to be changed at all, but they give you flexibility over exactly how the node communicates with other nodes on the network. In my case, since the reports were to be broadcast, they couldn't be acknowledged by all nodes receiving them.

I, therefore, changed the DLL Acknowledgement service from Acknowledged to Unacknowledged. Although CEBox's libraries automatically force any broadcast messages to use an Unacknowledged service, I did it to avoid any later confusion.

WRITING I/O ROUTINES

Once the Contexts, Objects, and IVs were chosen, I linked the hardware and the corresponding IVs. The current time was read from a battery-backed hardware clock, and this value updated the IVs in the Real Time Object.

The input-button values were also read, and if they corresponded to a change in the current time (i.e., incrementing the month, resetting the seconds, etc.), the hardware clock was updated. The LCD display was then updated twice a second.

Digital-clock manufacturers have most of this code written, so they only need to make small modifications to call functions within CELib that access the IVs.

However, new scenarios crop up with networked products, and I had to deal with these. For example, what if another node sent a message changing the current-time IV value?

If 1 wanted this to change the actual time, I couldn't blindly read the time from the hardware clock, since it would overwrite the current-time IV.

Instead, I had to see if current_time had changed since the last pass through the I/O routine, and if so, I then updated the hardware clock.

But, what if another node simply changed the value of the minute IV? I had to draw the line somewhere, so I decided to allow other nodes to change the current time but only through the current-time IV.

Since all the numeric IVs in the Real Time Object were updated with the reading from the hardware clock, changes in the numeric IVs due to messages from other nodes could be ignored since they would soon be overwritten.

However, to make it obvious to other nodes that their actions were in vain, all the numeric IVs in the Real Time Object were specified as read-only.

This attribute specification doesn't mean those IVs' values couldn't be altered from within the node-only

reporting_condition Expression	Hex
C	43
AND	E0
m	6D
DELTA	ED
5	35

Table *I-The report ing_condit ion IV tests to see if 5 min. have passed. Since I wanted to report the value of the current_time IV (and not minute), its label had to be inserted before the {'m' DELTA '5'} expression.*

that other nodes would be unable to change their values.

If another node used an Explicit_Invoke or Explicit_Retry service to try to change the value of one of the numeric IVs, a Failure packet with an error code of 15 (IV Read Only) is returned to the node.

WRITING MAIN()

As with most products using microcontrollers, no operating system is provided with CEBox.

Instead, as you can see in Listing 1, you structure your code around an infinite loop in the main() routine, calling any initialization code before the loop.

A call within the infinite loop to the CLBController function enables CELib to process incoming messages, send outgoing ones, verify all reporting conditions, and call each I/O routine.

CELib also uses two interrupts. An internal one increments various counters, while the transceiver uses an external one to indicate that it's ready to receive or transmit data.

report-header

Hex	Byte Description
05	Time Context
02	Object 2: Real Time Object
45	setValue Method
43	C: current-time IV
F5	Delimiter Token

Table 2—*The report_header IV directs the report to a specific IV in the destination node. In this case, it happens to be the same IV that I was reporting—the current_time Win the Real Time Object of the Time Context*

For my clock, there was little need to add to this routine. Besides the usual initialization code, the state of the input buttons within the infinite loop was verified to see whether the user requested one of CELib's address acquisition and propagation functions.

Briefly, these are procedures specified within the CEBus Standard that allow a node to either acquire a new address, acquire the system address from the master node, or become master node for 5 min. They are a simple way of setting a new device's system address to the same system address as other nodes in the house.

FILLING IN USR ROUTINES

USR routines are functions the user writes and CELib calls. They either provide feedback to the application or access the nonvolatile memory (NVM).

CEBox provides templates for these functions. Filling in feedback routines is purely optional, so I only filled the routine that provides feedback on the address-acquisition process. I used this to display messages on the LCD.

However, filling in NVM access functions is mandatory, since the user can implement NVM in any manner. The functions that needed coding were USRGetNvmByte (uint Nvm_Address) and USRSetNvmByte (uint Nvm_Address, uchar Byte).

The coding proved quite simple for the 256-byte I²C serial EEPROM on the target CPU board, especially since a generic I²C driver was already written and used to read from and write to the hardware clock.

COMPILING THE PROJECT

Once the code was written, it only took a couple steps to compile it into a

Listing 1—*CEBox doesn't include an operating system for the target* microcontroller. The manufacturer *simply places their initialization code at the beginning of the* main() routine and then structures *the rest of their code within an* infinite loop.

```
void main (void)
{
  CLBInitializeCELib();
  LcdInitialize();
  Code_Enabled_Flags |= GET_TIME_FROM_HARDWARE_CLOCK_BIT;
  while (TRUE) {
    CLBController ();
    switch (INPUT_BUTTONS_PORT){
      case BUTTONS_ACQUIRE_MASTER:
        LcdWriteRomStringFromLcdHome (ACQRING_MASTR);
        CLBAcquireMaster();
        break;
      case BUTTONS_ACQUIRE_NEW_ADDRESSES:
        LcdWriteRomStringFromLcdHome (ACQNG_NEW_ADDR);
        CLBAcquireNewAddresses();
        break;
      case BUTTONS_ACQUIRE_ADDRESSES_FROM_MASTER:
        LcdWriteRomStringFromLcdHome (ACQUIRING_TEMP);
        CLBAcquireAddressesFromMaster();
        break;
      case BUTTONS_FACTORY_RESET:
        LcdWriteRomStringFromLcdHome (RSET_TO_DEFAULTS);
        #pragma asm
          LJMP 0; branches to start of CSTART.ASM, resetting node
        #pragma endasm
          break;
    }  /* end switch */
  }/*  end  while  */
}/*  end  main  */
```

file that could be downloaded to the target microcontroller, the 80CE558.

First, I used CEBox's Project Manager to specify which source file to include in the executable. The dependencies were also specified for each file, so only those that needed to be recompiled would be after a small bug fix.

Clicking the Generate button in CEBox created link- and make-files based on the information entered into the Project Manager, as well as a batch file to execute them.

It then created a source file containing the Contexts, Objects, and IVs specified within the Context Builder, another source file containing the protocol parameters specified within CEBox, and a header file containing a constant that indicates up to which NVM address was reserved for CELib use.

Once this was done, I compiled and linked all these files into an executable using the BSO 805 1 tools (the only compiler currently supported) so I could see, among other things, which constants had been forgotten and which prototypes didn't match.

Once all the compiling and linking bugs were fixed, it was time to download the compiled clock to the CELab.

I connected the CELab to a serial port on the back of the computer and selected COM2 as my communications port within CEBox. Using the CELab Communications tool in CEBox, I downloaded the compiled .HEX executable to the CELab.

Clicking on Free Run started the clock, and for the most part, it worked on the first try! After fixing bugs in the display routines and in the day-of-week calculation, I recompiled the clock and downloaded it to CELab. A functional CEBus-compliant master clock was born.

ONLY A TICK AWAY

When all was said and done, it took surprisingly little work to create a CEBus-compliant clock that could keep all my other CEBus devices on time. Although several additional scenarios cropped up, most of the work concerned input-button processing and time and message display on the LCD.

Having my clock broadcast the time at 5-min. intervals was accomplished by initializing the values of three IVs. It took less than a minute!

Home automation has, until now, principally been the realm of expensive custom-designed systems.

However, the CEBus Standard is rapidly changing the face of the marketplace. Inexpensive, reliable, and compatible networked products will soon be available to one and all. ❑

Adrian Dunn works as a computer analyst for Domosys Corp. He is an Engineering Physics graduate from Queens' University, ON, Canada. You may reach Adrian at *adunn@domosys.corn.*

CONTACT

CEBus Industry Council (CIC)
4405 Massachusetts Ave.
Indianapolis, IN 46218
(317) 545-6243
Fax: (317) 5456237
cebus-staff@cebus.org
http://www.cebus.org/

SOURCES

BSO 8051 tools
Boston Systems Office/Tasking
Norfolk Place
333 Elm St.
Dedham, MA 02026-4530
(6 17) 320-9400
Fax: (617) 320-9212

CEBox, CELab, CEWay PL-III
Domosys Corp.
350 Rue Franquet, Ste. 110
Ste-Foy, PQ
Canada G1P4P3
(418) 658-9545
Fax: (418) 657-2123
info@domosys.com

CENode-CEThinx, P400 modules
Intellon Corp.
5100 W. Silver Springs Blvd.
Ocala, FL 34482
(352) 237-7416
Fax: (352) 237-7616
http://www.intellon.com/

I R S

401 Very Useful
402 Moderately Useful
403 Not Useful

XRaCS

FEATURE ARTICLE

The X-10 Radio-Control System

Chris Arndt

Want to control your house without the aggravations of a range-limited remote control or a phone-based system? Chris uses amateur-radio band frequencies and enjoys voice-synthesized feedback to commands.

While the X-10 power-line control system is a useful way to control lights and other devices around a home or business, there just isn't a handy way to operate X-10 controls from a distance.

The original system sold in the early '80s had a hand-held ultrasonic remote control. It only worked for line of sight in the same room as the control box, and most users spent a lot of time waving it around to get the command to take.

These days, you can buy a few different radio remote controls that fit in your hand or hang on a keychain. In my experience, their range is limited, the number of different units you can control is limited, and if you're trying to control something you can't see, you don't know if the base unit received the radio command. What it lacks is some sort of feedback.

I sort of solved this problem a while ago with an X-10 server I built out of an old Apple II+. I call it on the phone, and it answers in a synthesized voice. DTMF (touch tone) commands tell the computer which X-10 signals to send, and the computer responds with vocal feedback of the device's name or unit number and the control command sent down the AC line.

It's pretty cool, but it has its own limitations. It's not useful on a single phone line, unless you want to call from work to turn on the hot tub, don't have anything like an answering machine on the same line, and don't care if other callers hear the computer answer.

I used the Apple server again in our new house, but only after we installed a phone system ("Do-It-Yourself Brain [Room] Surgery" *INK* 74). The X-10 server is on a dedicated extension of the phone system, so an X-10 control console is as close as the nearest phone.

Hooking up this server improved X-10 control around the house, but I wanted something that combined the convenience of the wireless ultrasonic or radio controls with the voice feedback of the computer control, had more range, and could control more X-10 modules.

A cordless phone works with the phone-based controller, but it still suffers from limited range. Constantly moving the handset from ear to view as 1 push buttons and listen to the computer is a pain. And, it's not suitable if you have a single line and no phone system.

I needed a better hand-held unit with some range, a DTMF pad, and a speaker rather than an earpiece.

NEW X-IO REMOTE CONTROL

As an amateur radio operator, I already had the basic field controller at hand as it were (see Photo 1). Almost all modern hand-held VHF or UHF ham radio transceivers have a built-in 16-button DTMF pad on the front.

All I needed was the equivalent of my Apple computer, the modem for DTMF detection and X-10 interface, and a speech synthesizer all connected to a radio on the other end. It didn't seem like too much to ask.

When I discovered that Parallax's Basic Stamp 2 had an X-10 output command, I knew the hardest part—the X-10 signal generation-was over.

In the Apple system, machine-language subroutines handle the X-10 zero-crossing detection and command generation. I had some DTMF receiver chips left over from my Apple project, and I scrounged and retuned a UHF Motorola telemetry radio to an appropriate portion of the amateur 440 band.

The Basic Stamp 2 has 16 I/O pins. I had a feeling I'd need most of them.

DTMF RECEIVER

As Figure 1 shows, the DTMF receiver is a Teltone M-957. It requires a color-burst crystal, a couple of capacitors for bypassing and DC blocking on

the input, 5-12 VDC, and an audio source.

It outputs the DTMF digit as a 4-bit parallel nibble and a separate valid-data strobe. It can be set to decode only the **12** most common DTMF digits (i.e., 0–9, *, and #) or all **16** (i.e., the common 12, plus A, B, C, and D).

The DTMF receiver uses five pins of the Stamp. I strapped the chip for **16** digits by grounding the 12/16 pin and for minimum sensitivity by grounding the A and B pins.

The Stamp's XOUT X-10 command uses two I/O pins and connects to an X-10 TW523 interface that plugs into any 120-VAC outlet. While the TW523 looks like a wall-wart power supply, its only function is to safely send the zero-crossing signal of the AC line to the controlling X-10 device (i.e., the Stamp) and safely couple the 120-kHz X-10 signal onto the power line.

RADIO TRANSCEIVER

The radio transceiver is a Motorola Rnet 450 originally used as a telemetry radio in the 450-MHz band. I purchased new crystals for it and retuned it for the 70-cm amateur band.

This unit runs on 12 VDC. Its transmitter has a nominal output power of 2 W-just about right for this project. However, I wouldn't necessarily choose this transceiver if I was shopping for a new one.

New, it's a bit expensive for this kind of project. Other manufacturers have better deals on equivalent units, if the application is commercial.

For amateur-radio use, this project could be adapted to almost any handheld, mobile, or base transceiver where this type of auxiliary operation is allowed by FCC rules. But, I got a good deal on a used transceiver, and it works fine. One bonus is its small size. The entire transceiver is contained in a 3.75" x 2.75" x 1.75" housing.

Keying the radio transmitter requires grounding the PTT (push to talk) line of the transceiver, which takes another pin. A MOSFET amplifies the output of the Stamp pin to the level necessary to activate the transmitter.

Telling the Stamp when to expect digits is more involved. It could monitor the valid-digit pin on the DTMF receiver, but it lacks security.

The radio has a carrier-detect pin that changes state when the receiver hears an on-frequency radio transmission. Using this as an activity indicator is worse than the valid digit pin, and its operation on this particular transceiver was erratic anyway.

TONE CODED SQUELCH?

One way different radio users share a common frequency without having to listen to each other's radio conversations is through Continuous Tone Coded Subaudible Squelch (CTCSS).

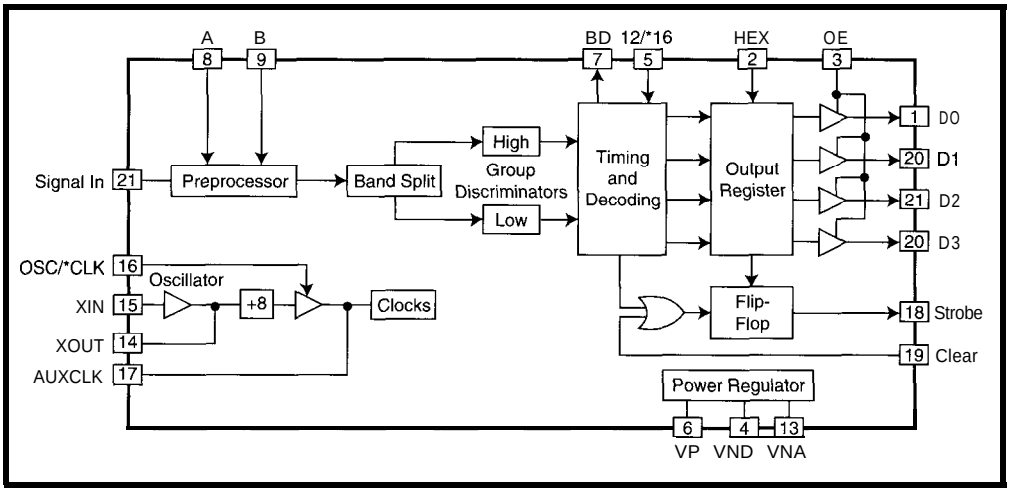


Figure 1--The *Teltone* M-957 contains *almost all the circuitry necessary to decode DTMF signals. The only external parts needed are a bypass capacitor and a 3.58-MHz crystal.*

CTCSS, sometimes known as PL because of Motorola's Private Line trademark, impresses a selected audio tone onto the transmitted audio.

The tones are in the range of 67.0-203.5 Hz, which is lower in frequency than the receiver's ~300-Hz low-frequency audio cutoff. A tone detector in the receiver prior to the audio filters listens for the correct tone and keeps the speaker muted unless the right tone is detected.

This way, several different users (e.g., a plumber, an electrician, and a carpenter), all with different CTCSS tones, can share a common radio channel without hearing each other's conversations.

Add-on encoder/decoder boards are available for radios that don't already have one. My radio didn't, and I realized that an add-on CTCSS decoder would provide a little security and a solid indication to the Stamp to look for incoming DTMF data.

Adding one to the Motorola unit required a little schematic investigation. I needed to connect it to the right part of the receiver so received tones would be present.

Usually, a CTCSS decoder connects at the output of the demodulator. In an FM receiver, it's the discriminator output, before any audio filtering.

The decoder I chose is the Communications Specialists TS-32P. It's a 1.25" x 2.0" board, runs on 9-18 V, and has a DIP switch to select the desired CTCSS tone.

It has several control output options. An audio output can be connected to the transmitter audio input for adding the selected CTCSS tone to a transmitted signal.

I connected the TS-32P's input to the radio transceiver's discriminator and the tone output to the modulator. That way, outgoing transmissions are encoded, and CTCSS decode can be used at the operator's end, too. I used the low-going detected-tone option to connect to the Stamp.

WHAT ABOUT VOICE?

Now, I have the Basic Stamp 2 as a controller, a DTMF chip for input, the TW523 for X-10 output, and the radio and CTCSS board for radio reception and transmitting. What's missing? Voice feedback!

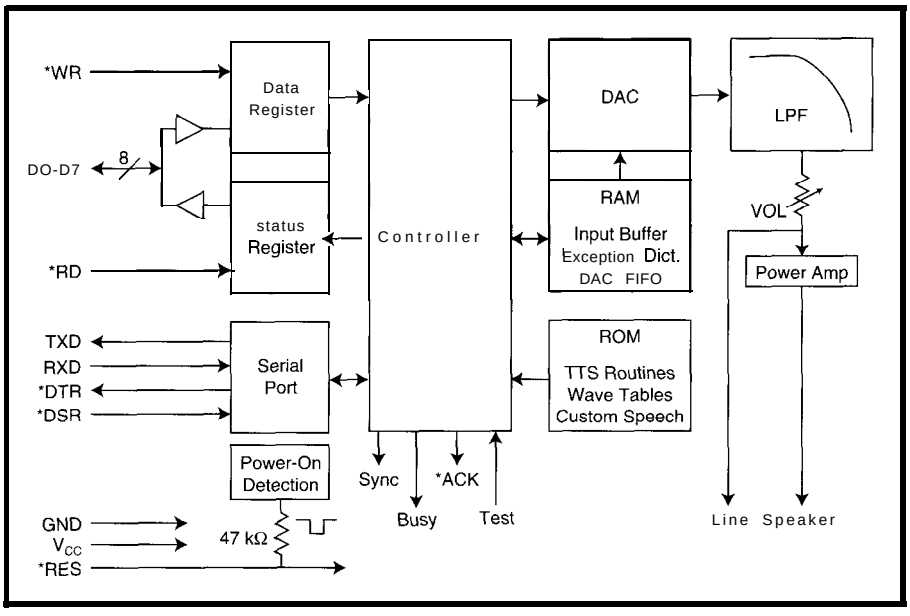


Figure 2—*TheV8600* is a complete text-to-speech *processor that accepts parallel or serial input and has line-level and amplified outputs. Its pronunciation rules can be* modified by *the user for* personal *preference or to speak a* different language.

Although I first chose National's Digitalker chipset, the hardware and software interfacing wasn't as easy as I'd hoped. It was going to take a lot of Stamp pins [maybe more than I could afford), and the vocabulary was limited.

In the back of an ***INK*** issue, I saw an ad for a single-board text-to-speech generator. Shown in Figure 2, the RC Systems V8600 takes TTL serial, parallel, or bus ASCII text input, and it outputs an audio-synthesized voice.

It can add inflection, has adjustable levels of digit and punctuation interpretation, and even has an onboard audio amplifier to drive a local speaker. It was just what I needed. It was more expensive than the Digitalker, but its ease of use more than made up for it.

To save Stamp pins, I used the serial-input option. This takes another Stamp pin to output the text to the board and one more from the SYNC pin on the speech board back to the Stamp, so the Stamp knows when the board is done talking. The SYNC pin goes high as the board starts speaking and goes low when it's done.

Figure 3 shows the hardware assembly. It fits on a 7″ x 9″ aluminum panel mounted in an 8" x 10" x 4" steel cabinet. I used adhesive-backed hook-and-loop tape to mount the CTCSS board and the speech synthesizer.

THE PROGRAM

The Stamp program is divided into two main sections and several subroutines (see the Circuit Cellar BBS or Web site). The first section performs setup and testing of the hardware.

It initializes the speech board, setting the communications baud rate and slowing down the spoken-text speed. The radio then transmits a test to confirm the transmitter works and the voice card is set up properly.

Next, a test X-10 command is sent. Since I use a different house code than the ones we use at home, it doesn't interfere with any other modules.

The X-10 test sends a House Code A Unit 10 N, pauses a second, and sends an OFF command. A small incandescent light plugged into an X-10 lamp module set to house code A unit 1 near the controller indicates whether the X-10 system is working.

The XOUT (X-10) command requires two pins on the Stamp-one to detect the zero crossing of the AC line voltage and one to send the X-10 signals to the TW523 for the power-line interface.

To properly time the sending of X-10 commands, the Stamp waits for the zero-crossing signal from the interface. If the Stamp isn't connected to the TW523 interface or the interface isn't plugged into a live AC outlet, the program hangs.

To keep this from happening during initial startup or after a hardware reset, the program uses the Stamp BASIC PULSIN command to look for the AC zero-crossing signal. If it's not there, it jumps to a routine that announces the fact over the radio.

Once initialization completes, the program enters its main loop-the supervisor. It looks for low on the Stamp pin connected to the CTCSS board, indicating that a valid subaudible tone was received.

When this occurs, the program goes to the X-10 routine. The subaudible tone isn't checked again (except in the transmitter keying subroutine) until it returns to the supervisor routine. The initial tone detection acts as a key to unlock the X-10 features.

The supervisor routine lets you add other features to the program, without rewriting major portions of it.

A complete X-10 command in this program consists of a single digit for the house code, two digits (1-16) for the unit code, and a DTMF letter (A or B) for an action command.

A is for On, B is for Off, and C (Brighten) and D (Dim) are reserved for a future enhancement. The DTMF "*" is a universal reset character if you lose your place. The "#" is unused.

This three-digit address scheme for a given X-10 module means that 160 module addresses out of 256 total are available by radio control, if the programming for all the house codes is added (10 digits available for house codes on a DTMF keypad x 16 module codes = 160 module addresses). At this point, I only need 32.

After the unit-code digit is sent, the Stamp replies by voice over the radio with the equivalent house-code letter. After the unit code, it echoes the number, which is then echoed over the radio. The program then loops back to the supervisor routine.

The X-10 routine consists of calls to a DTMF digit-grabbing subroutine and actions taken on those digits. The first digit is looked up as a house code.

To keep consistency with the old Apple program still running on the phone system, a DTMF 1 represents house code K and 2 equals house code L. Since house code A was already declared and used in the Stamp program, I equated it to 3.

Once a complete X-10 sequence is sent, the program jumps to the identification routine and loops back to the supervisor.

TRANSMIT AND TALK

I couldn't figure out an easy way to pass text strings between program parts, so I split the transmitting and speaking subroutine. The two parts are called before and after sending the serial message text to the speech board.

The subroutine KEY waits for the CTCSS I/O pin to go high, indicating that the radio of the person sending commands has stopped transmitting and is ready to receive. It turns on the transmitter and pauses so the CTCSS decoder on the other end can open the audio on that receiver.

The KEY subroutine returns, the message text is sent to the speech

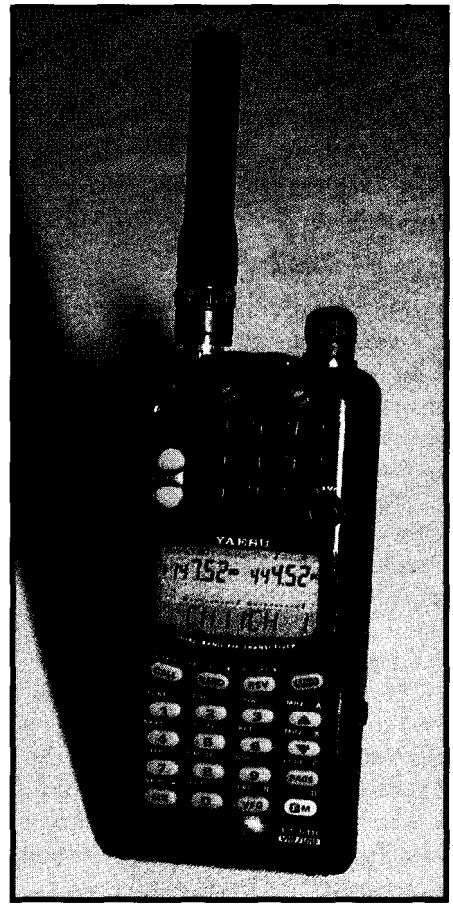


Photo I--The *Yaesu FT-51R* is a dual-band *144-/440-MHz* hand-held transceiver. It has a full *16-button DTMF* keypad and CTCSS encode and decode, both important for operating *XRaCS*.

board, and DKEY is called. The speech board raises the SYNC pin to indicate that vocalization has started.

D K E Y waits for this pin to go high and then drops when speech ends. The transmitter is deenergized, and the program continues.

GLUE STUFF

Most connections between the subsystems are direct, and you don't need a lot of glue to stick it together and make it work. The transmitter keying requires a MOSFET to match the Stamp to the PTT line of the radio.

The DTMF receiver needs two capacitors and a color-burst crystal. During development, I added an LED indicator to the CTCSS decoder output and another to the PTT driver to visually track program execution.

The radio and CTCSS board work on 12 VDC. I added a 5-V regulator for the V8600 speech synthesizer and the Stamp carrier board. The speech board has an onboard amplifier, so I added a miniature phone jack. Now, I can plug in a speaker and listen during testing.

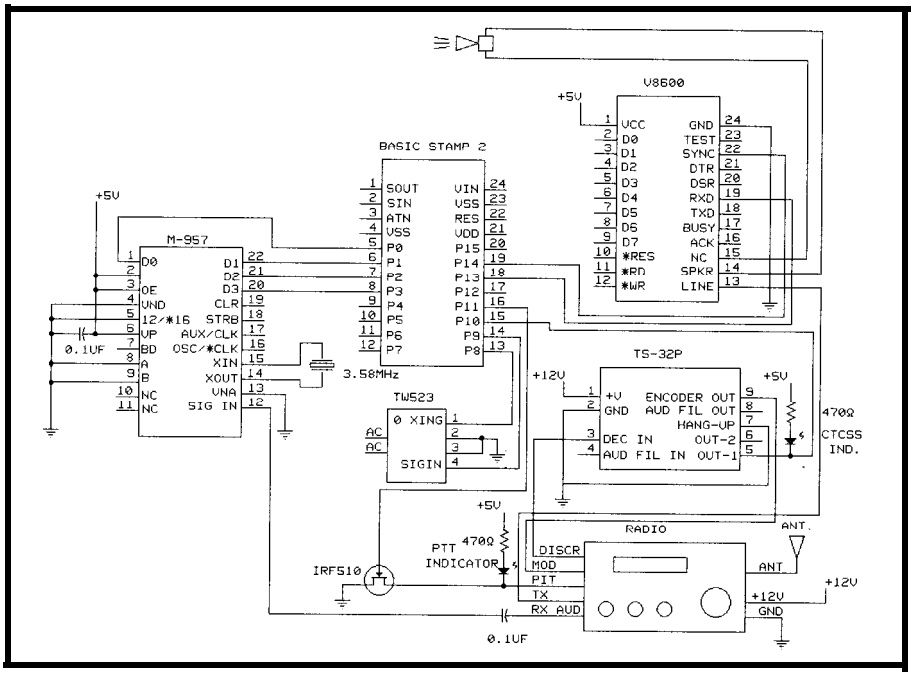


Figure 3—*The XRAGS is assembled from five main subsystems-the Basic Stamp 2, the DTMF decoder, the radio, the CTCSS board, and the voice-synthesis board. Mating these systems requires few extra parts.*

IDENTIFICATION

The FCC requires radio amateurs to identify themselves once every 10 min. and at signoff. Since this is used in the amateur service, I had to keep it legal.

There are published Morse Code routines for Basic Stamps. But, the voice synthesizer makes it easier to have my callsign spoken.

I didn't include a clock or timer. As a low-use device, it's easiest to transmit the identification after every complete X-10 sequence.

SAFEGUARDSAND PRECAUTIONS

The CTCSS adds some security from casual hacking of the system. A DTMF password adds more. The radio transceiver has an integral time-out timer to kill the transmitter if the program locks up with the transmitter keyed.

Even though there's audio feedback that the command is sent down the power line, there is no feedback that it actually takes effect. A failure here might be caused by interference or collision on the power line with another X-10 command.

PROJECT WRAPUP

I hope this article serves more as a catalyst than a project to duplicate. I only touched briefly on the individual components. Any of the major subsystems (i.e., radio, CTCSS, DTMF decoder, Basic Stamp, V8600, or X-IO) could be separate articles.

The radio can be substituted or omitted and the controller added to an existing system. If I hadn't scrounged

this particular Motorola unit, I'd have used something else. It's limited by the carrier-detect action and can cost more than all the rest of the parts.

With the elimination of the CTCSS board, this unit could be built to plug into the speaker-mike jack of most commercial or amateur hand-held or base transceivers, using the radio's internal CTCSS decoder and the valid-digit pin on Teltone's DTMF receiver. The speech board can be substituted or even eliminated, if you can live with beep-boop feedback.

My RC Systems speech board has a wide range of programmable intonation and inflection. Right now, "he" sounds really bored. I need to work on the board initialization and message punctuation to liven him up a little.

This project took 11 of the Stamp's 16 pins. I had a serial LCD display on one pin during debugging. If I put it back in, I could display recently sent commands. Or, I could use those pins for critical feedback about the status of the AC device controlled by an X-10 module.

The spare pins could monitor some switches or serial ADCs. Any change could be announced by radio and/or used to activate an X-10 device. It would be great for intrusion alerts, low temperature alarms, or similar needs.

You can increase security by using the Stamp to control the CTCSS decoder board. The frequency selector on the board is a DIP switch. Leaving one or more switches open and using pins on the Stamp to control those inputs may allow for some sort of remote or rolling CTCSS tone change and decrease a hacker's chance of getting in.

A serial real-time clock could add time-of-day event scheduling or time announcements. Or, I could increase security by changing day to change CTCSS or to change an added DTMF password.

So many dreams, so little time. ❑

I'd like to thank Jeff Martin of Parallax and Jon Williams for their comments, suggestions, and help with a couple sticky parts in the program.

Chris Amdt, KD6DSI, has been a licensed amateur since 1991. He builds his projects when he's not working as the Telemetry and Instrumentation *Technician for the City of San Luis Obispo, CA. You may reach Chris at carndt@slonet.org.*

SOFTWARE

Complete source code for this article is available on the Circuit Cellar BBS or Web site.

SOURCES

Basic Stamp 2
Parallax, Inc.
3805 Atherton Rd.
Rocklin, CA 95765
(9 16) 624-8003/8333
Fax: (916) 624-8003

Motorola RNET 450 transceiver
Motorola, Inc.
Paging Products Gr.
3301 Quantum Blvd.
Boynton Beach, FL 33426
(561) 739-3880
Fax: (561) 739-3815

TS-32P CTCSS decoder/encoder
Communications Specialists, Inc.
428 W. Taft Ave.
Orange, CA 92665
(714) 998-3031
Fax: (714) 974-3420

Teltone M-957 DTMF receiver
Teltone Corp.
22121 20th Ave. SE
Bothell, WA 98021
(206) 487-1515
Fax: (206) 487-2288

V8600 text-to-speech board
RC Systems
1609 England Ave.
Everett, WA 98203
(206) 3553800
Fax: (206) 355-1098

X-10 **TW523** interface
Home Automation Systems, Inc.
151 Kalmus Dr., Ste. L4
Costa Mesa, CA 92626
(714) 708-0610
Fax: (714) 708-0610

IRS

404 Very Useful
405 Moderately Useful
406 Not Useful

FEATURE ARTICLE

John Morley

HCS II-Compatible Automated Lawn-Irrigation System

Summertime is coming and with it, the hot dry drought. However, just because the rain clouds aren't complying is no reason for your lawn to turn brown. John tells you how to keep green.

I don't think anyone would argue about the beauty of a lush green lawn. It's not only eye-catching, but it also increases your home's value.

And, acquiring the perfect lawn is easier than you might think. To get a healthy, thick, green lawn, you must ensure that it receives enough sunlight, fertilizer, and water.

Of course, you can take lawn care to an extreme. For example, my neighbor periodically color-codes the problem areas of his lawn with golf tees. Red means the lawn needs more water, purple calls for more fertilizer, and blue beckons more grass seed.

But, like most people, you probably don't have a lot of time for your lawn. You need the best lawn possible with the least amount of effort.

When it comes to keeping a healthy lawn, with the exception of mowing, you probably spend more time watering your lawn than anything else. Obviously, automating this time-consuming and messy job would be useful.

An automated lawn-irrigation system provides the proper amount of water each day and, as a bonus, delivers exactly the right amount just where you want it. It dramatically improves watering efficiency, so you see more improvement from the water you use.

I decided to install a lawn-irrigation system following the summer of 1995. That summer was one of the driest on record in New England. I fought a never-ending (mostly losing) battle to keep my yard green.

Naturally, as a home-automation enthusiast, I wanted to use an HCS II interface for the sprinkler system. In this article, I describe the design and installation of an automatic lawn-irrigation system and an HCS II sprinkler controller.

SYSTEM OVERVIEW

Automated lawn-irrigation systems generally consist of an underground network of plastic pipes supplying water to water-sprinkler spray heads strategically placed around the yard. A typical system is segregated into zones, each comprising three or more spray heads.

Each zone connects to the main water supply by electrically operated water-control valves that enable each zone to be independently controlled.

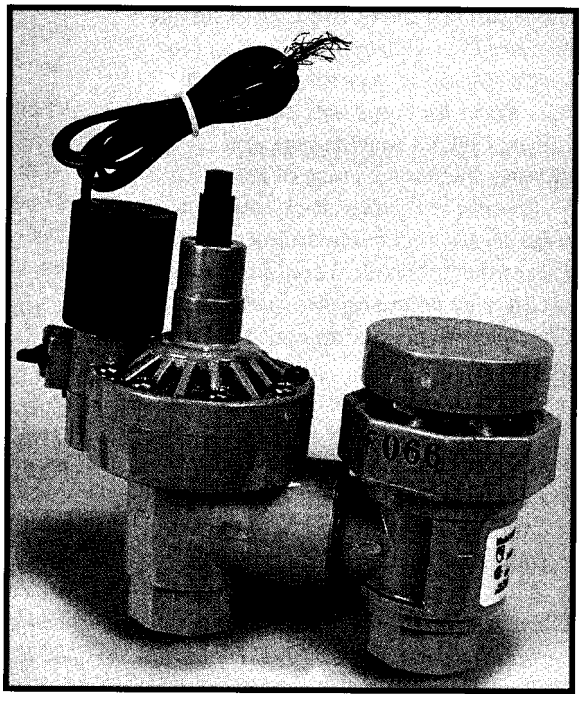


Photo 1—*Valves* of this type control the flow of water to the spray *heads in your irrigation system.*

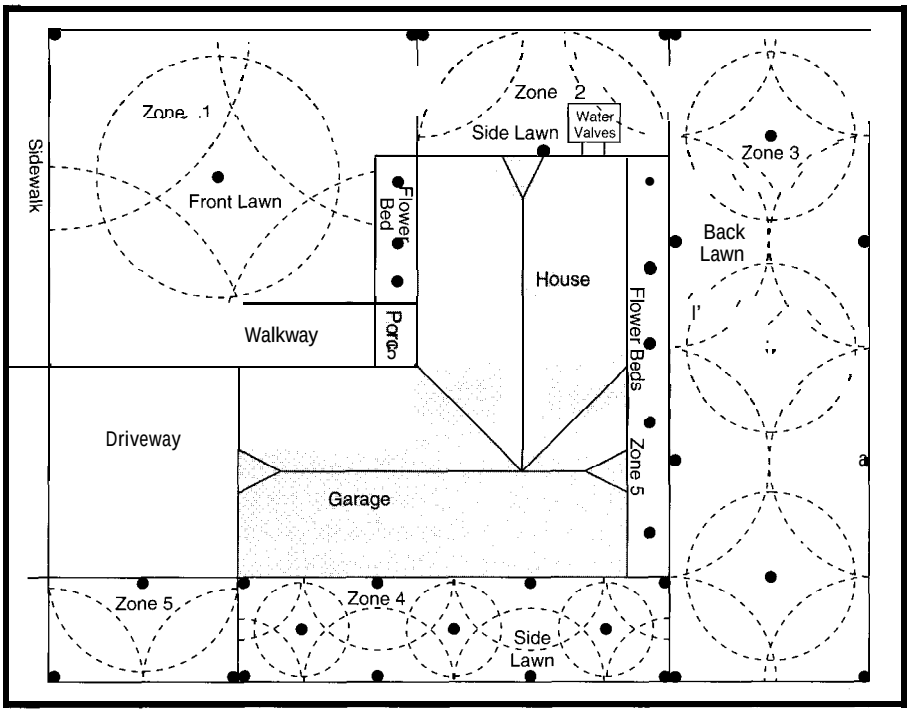


Figure 1--This sample *lawn-sprinkler* design layout shows *all* of *the components of an* irrigation system.

Most systems incorporate an electronic timing mechanism to automatically sequence the zones beginning at a specific time each day. Many also include moisture-sensing devices that disable the system when no more water is needed.

The many different types of spray heads let you customize an installation to your yard's size and layout. Most hardware superstores stock a wide variety of do-it-yourself lawn-irrigation supplies at reasonable prices. 1'11 describe the most common components.

A water-pressure gauge measures the static water pressure of the main water-supply line. This pressure reading determines the size of the water-distribution pipe and the number of spray heads each zone can accommodate.

A backflow-prevention valve keeps debris and impurities from contaminating your household water supply. It should be located as close to the main water-supply source as possible.

Most local plumbing codes require this, but use one anyway, even if it's not required. Include an on/off (ball) valve so you can turn the system off for maintenance or in cold weather.

The underground pipe connecting the spray heads and water-control valves to the main water supply is generally a rigid black PVC or polyethylene plastic tubing. The tubing is usually 0.75" or 1" I.D., depending on the static water pressure of the water-supply main and the number of heads per zone that you're using.

Electric water-control valves, depicted in Photo 1, let you remotely control the system's operation via an electronic timer or home-automation interface. With multiple valves, you can segregate your irrigation system into independent operating zones.

The primary reason for zones is that the total number of spray heads exceeds the flow rate and static water pressure that the main water supply can sustain.

Most water-control valves operate at 24 VAC, which is provided by the electronic timer or home-automation interface. Some valves also have internal backflow-prevention valves. They are typically buried inside a plastic enclosure or vault to protect them from direct soil contact.

Automatic drain valves, which are installed at the lowest points of the irrigation system, let water escape if no water pressure is present. When the irrigation system isn't operating, standing water still drains, preventing damage due to freezing.

This device makes winterizing your irrigation system as simple as turning off the main water-supply valve. Some component manufacturers recommend purging the system with compressed air if your region undergoes prolonged periods of freezing temperatures.

Rain and soil-moisture detectors inhibit the automatic system when watering is not needed. The rain detector mounts above ground [typically, it's screwed to the side of the house), while the soil-moisture detector is buried a few inches underground.

Each device contains a single-pole switch that's normally closed. These detectors contain an absorptive material that expands when exposed to water, activating an internal switch.

They can be wired in series with the common ground return for all of the individual water-control zone valves. When rain or soil moisture is detected, the switches open, disabling the system zone valves.

In a home-automation installation, connect the detectors to a digital input on the home controller. They will signal the home-automation system that no more watering is needed.

Fertilizer injectors automatically deliver liquid fertilizer from a reservoir connected in line with the main water supply feeding the system. Most models can be adjusted to control the water/fertilizer mixing ratio.

Injectors are available in many sizes. Choose one based on your system's flow rate. Since this device applies fertilizer to your lawn in a consistent and uniform manner with virtually no effort on your part (you must keep the reservoir filled), the days of pushing a hand lawn spreader are over.

Fixed and pop-up spray heads are the simplest and cheapest way to get wide-area coverage. The fixed head mounts 2–6" above ground, so it's often placed near the lawn's perimeter where it doesn't interfere with maintenance operations.

A pop-up head, as shown in Photo 2, is mounted underground but extends 2-6" above ground level when operating. Because it is otherwise below ground level, it can be located anywhere on the lawn.

Either head type can produce a water pattern ranging from a narrow stream to a full circle. Adjustable models let

Listing I--This *XPRESS code controls the lawn-sprinkler controller. The lawn is watered at 6:00 A.M. and 3:00 P.M. each day. First, the status of the rain and soil-moisture sensor is checked. A read to the lawn-sprinkler module returns a 1 if watering is required and 0 otherwise. The sprinklers are activated by sending a value of 25.5 to the controller.*

```
! Check lawn status at 6:00 A.M., and water if necessary.
IF (TIME>=06:00:DY AND NETBYTE(7)=1) THEN
  NETBYTE(7)=255
END

! Check lawn status at 3:00 P.M., and water if necessary.
IF (TIME>=15:00:DY and NETBYTE(7)=1) THEN
  NETBYTE(7)=255
END
```

you water **odd-shaped areas.** They typically have an operating radius of 20 and distribute **l-3** gal. per minute and are best suited for small to moderately sized open areas.

Soaker hoses are semiporous hoses constructed of a pliable foam rubber that lets water seep out at slow, controlled rates. They can be placed directly on the ground or buried just beneath the surface. These efficient devices deliver water directly to the soil around the roots, reducing evaporation and the amount of water needed.

Low-volume drip nozzles provide a continuous water drip directly to the roots of individual plants. Spray nozzles offer a fine atomized mist for the leaves of even the most fragile plant life. These nozzles are extremely low volume and can even be used indoors.

IRRIGATION-SYSTEM DESIGN

By far, the most important aspect of the system's installation is your initial planning. Be thorough. Complete your plan before the first dirt is shoveled.

You must first contact your local water and building departments to find out what, if any, plumbing or building codes apply. Some areas also require permits.

As well, see what type of backflow-prevention device you need and where you should install it. You don't need a common backflow valve if all the electric zone valves have them built in.

Contact your local utilities to determine the location of underground utilities. Exposing or damaging any utility should be strictly avoided.

Using a water-pressure gauge, measure the static water pressure of your domestic water supply at the outdoor faucet where you connect the garden hose. The threads on the gauge are specifically designed to fit the threads on the faucet.

At the same time, measure the size of the main water-supply line connecting to the irrigation system by wrapping string around the outside of the pipe. Remember, a circle's circumference (c) (i.e., the measured string) is related to its diameter (d) by π:

$$d = \frac{c}{\pi}$$

The standard convention for categorizing copper tubing is to use the I.D.—not the O.D.-of the pipe. Therefore, don't be surprised when your calculations return values slightly greater than 0.75 (0.75" pipe), and 1.0 (1" pipe). The value of the static water pressure and the diameter of the pipe determine the available flow rate for your system.

The final planning step is perhaps the most important. Draw a scale diagram showing your property's perimeter and the boundaries of all buildings, garages, and sheds. Include sidewalks, patios, driveways, fences, and walls. Finally, note all trees, shrubs, gardens, lawn areas, and the location of the water supply.

This diagram is the master plan for laying out and positioning all system components. If you adequately plan the layout, you ensure complete watering coverage.

To begin, select the spray heads that provide the best coverage with the least overlap. Review manufacturers' specifications for which spray heads are available.

Next, calculate the combined flow rate of the spray heads. The total num-

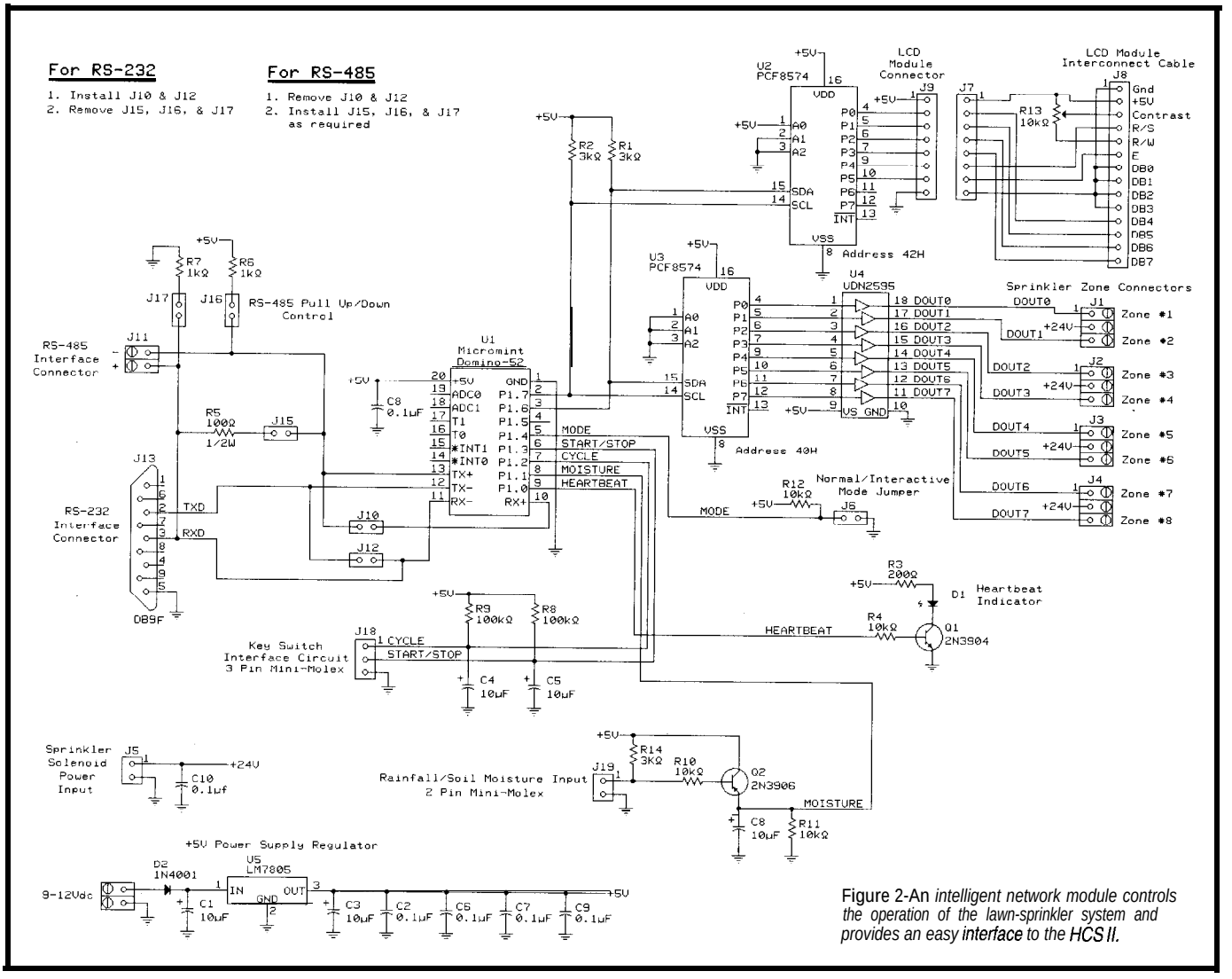


Figure 2-An *intelligent network module controls the operation of the lawn-sprinkler system and provides an easy interface to the HCS II.*

ber of zones required by your system equals the combined flow rate for all spray heads divided by the maximum flow rate of your water supply.

Always round up the result to the nearest whole integer to avoid designing a system without enough flow capacity. Group the spray heads to form the appropriate number of system zones. The total flow rate for any zone must not exceed the available flow rate of the main water supply.

There are several piping methods for supplying water to individual zones. The first method groups all of the water-control zone valves connected to a common water-supply distribution manifold. The valves are located underground close to the water-supply main, and individual pipes are run to supply each zone.

This method lets you conveniently locate all the water-control valves in one underground vault, while running a minimum amount of control wiring. Because there are no common distribution pipes, a lot of pipe is required.

The sample design layout in Figure 1 was condensed from my own experience and the layout guides provided by several irrigation-system component manufacturers. Get your own copies of installation guides and use them.

HCS II SPRINKLER INTERFACE

lawn-irrigation system and my HCS II, I had to interface it to my supervisory controller. I quickly dismissed the idea of directly wiring the sprinkler zone valves to the HCS II as it required running too much wire.

Instead, I chose to control the zone valves via an RS-485 network-connected DIO port in my garden shed. The controller was physically close to the valves it would control, protected from the elements, and easily accessible for manual control.

Since I wanted to add manual-control capability to the controller, I built my own network module rather than buying an off-the-shelf solution (e.g., the HCS II DIO-Link). I needed a programmable module, so I picked Micromint's Domino-52 microcomputer/controller module.

Domino contains an 80C52 controller with an onboard BASIC interpreter, 32 KB of SRAM, 32 KB of nonvolatile EEPROM, 12 parallel I/O lines, and drivers for RS-422/-485 and RS-232A serial communications-all in a package occupying less than 0.8 in:?.

To maintain compatibility with the HCS II, my network module is programmed to mimic a subset of the DIO-Link command set. The sprinkler controller, therefore, appears to the

HCS II as a network-connected DIO-Link.

The commands that this controller is programmed to accept are:

- SDP=34 Set Port to Hex 34 (byte)
  This command allows the HCS II to send a complete digital word (hex byte] to the lawn-sprinkler controller. The controller can interpret this word and initiate any number of activities (e.g., a complete watering sequence or the activation of a single zone).

- QDP Returns Whole Port Value (byte)
  This command enables the HCS II to receive a complete digital word (hex byte) from the sprinkler controller. The HCS II can request that this data be sent at any time. This command determines the status of the rain and soil-moisture sensor.

To test the status of the rain and soil-moisture sensor, a read from the controller module is initiated as shown in Listing 1. A zero is returned if watering is not required, and a one is returned otherwise.

If the water-sensor test returns one, a watering cycle is initiated by sending the controller a value of 255. Once started, the sprinkler controller activates each available zone in sequence for IO-min. periods.

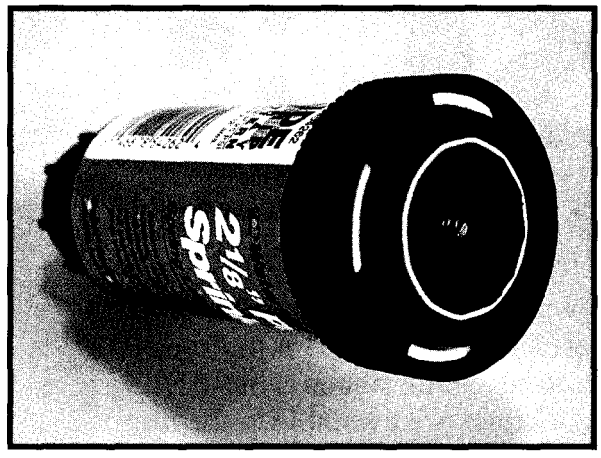


Photo 2-A pop-up spray head *is* mounted *underground but extends 2–6″ above ground during operation. Many are water-pressure activated, and gravify causes them to retract*

The prototype sprinkler controller is shown in Photo 3.

INTERFACE CIRCUIT DESCRIPTION

As Figure 2 shows, the HCS II sprinkler-controller is based on the Domino-52 microcontroller. The interface has an onboard +5-V voltage regulator (U5) to provide a stable source of power for the Domino, an LCD display, and a number of support ICs. The recommended power input to the interface board is +9 to + 12 V.

Communication with Domino [for both programming and operation] is via a serial communications port configured for either RS-232A (O-5 V only) or RS-485 (HCS II network compatible). J10 and J12 select either RS-232A or RS-485 serial port mode.

When Domino is configured for RS-485, J15, J16, and J17 select the proper network termination. Consult the HCS II network operations manual for information on proper termination.

For RS-232A communications mode, J10 and J12 should be installed, while J15, J16, and J17 should be removed. For RS-485 communications mode, J10 and J12 should be removed, while J15, J16, and J17 should be installed.

A 20-character x 2-line alphanumeric LCD interfaces to the Domino via a Philips PCF8574 I/O expander IC (U2), which connects to the Domino via an I²C bus.

A second PCF8574 I/O expander (U3) provides an eight-bit digital output port that controls up to eight sprinkler-zone valves. An Allegro UDN2595 output driver (U4) provides high-current capability to the output port and lets the zone valves connect directly to the port.

I made a number of connections to Domino's dedicated DIO bits. Port 1, bit 0 is configured as an output and provides control for an LED that flashes continuously when power is applied, signaling that the program is executing.

Port 1, bit 1 is configured as an input and reserved for the rain and soil-moisture sensor, which connects to the interface board at J19.

The output of the rain and soil-moisture circuit connects directly to a dedicated I/O bit on Domino itself.

Port 1, bits 2 and 3, are configured as inputs and provided as switch inputs. These bits read the status of push-button switches located on the front panel of the interface.

Port 1, bit 4 is configured as an input and enables the interface to read the status of a mode configuration jumper.

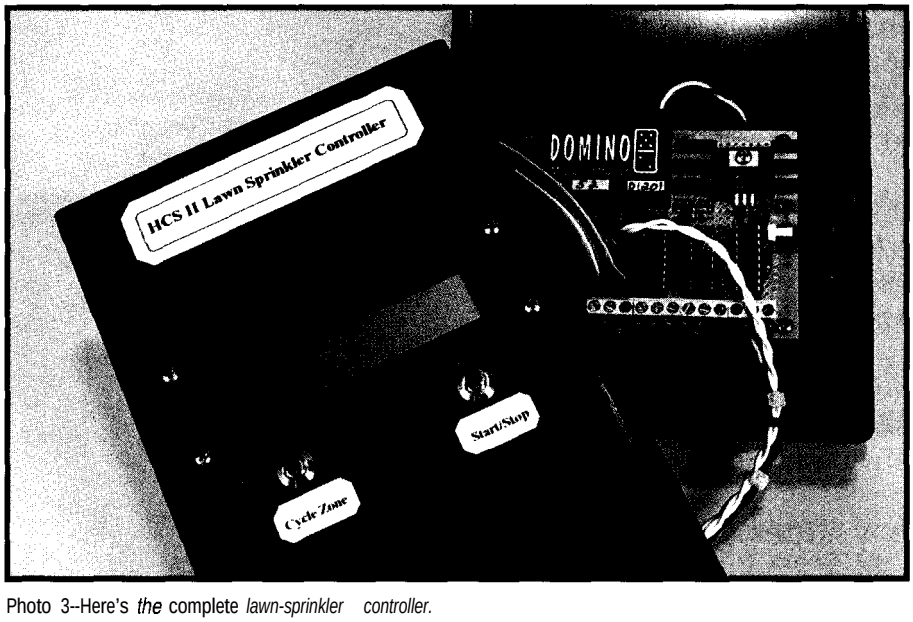


Photo 3–Here's *the* complete *lawn-sprinkler controller.*

CONTROLLER INSTALLATION

Begin installation by digging holes for each spray head, water-control valves, automatic drains, and outdoor

water faucets. Only remove enough dirt for the spray head to be buried so its top is flush with ground level.

When it comes to burying the water-distribution pipe, there's only one way to go! Rent a commercial trenching/pipe-burying machine. You can do a faster, neater job than by hand.

A typical trenching machine has large pneumatic tires and is powered by an 8–10-hp gasoline engine. As the machine moves over the ground, it slices a narrow opening in the soil with a trench-cutting blade. Some models [e.g., DitchWitch] bury the pipe as the machine progresses.

Connect all the components to the water-distribution pipe. Next, install the rain or soil-moisture detector, following the manufacturer's instructions. Finally, connect the water-control valve electrical wiring to the controller. A small wall-mount transformer provides power for the control valves, which need 24 VAC.

Once your system is installed and running, immediately fertilize the lawn several times to repair any damage caused during installation. I couldn't detect any trace of installation after about three weeks.

FREE TIME

My ideal home-automation installation is a system that consistently and reliably performs mundane and repetitive tasks. Lowering the thermostat when the house is unoccupied or turning on lights at dusk exemplify how home automation simplifies things.

Sometimes, we still benefit by automating activities we do less often. When I add a feature or capability to my HCS II, I begin by evaluating its utility.

A lawn-irrigation system, controlled by my HCS II, represented an enormous long-term benefit. Since a lawn needs frequent watering during the hot, dry summer, I spent l-2 h per day, several times per week, manually watering the lawn. It was a huge time commitment.

On an absolute scale, not only is my lawn fuller, healthier, and greener, but I now have time to lay back in my hammock and ponder the really *important* home-automation projects-like controlling the hot tub. ❑

John Morley is the senior electrical engineer for a small Boston-area manufacturer of custom electronic test equipment. He designs instrumentation used to measure the thermal properties of packaged semiconductor devices and the reliability of passive electrical interconnects. You may reach John at endeavor@ultranet.com.

SOURCES

Domino-52
Micromint, Inc.
4 Park St.
Vernon, CT 06066
(860) 871-6170
Fax: (860) 872-2204

PCF8574 I/O Expander
Philips Semiconductor
811 E. Arques Ave.
Sunnyvale, CA 94088
(408) 991-2000
Fax: (408) 991-3581

UDN2595 **output driver**
Allegro Microsystems
115 Northeast Cutoff, Box 15036
Worcester, MA 0 16 15
(508) 853-5000
Fax: (508) 856-7434

Lawn-Irrigation Parts
The Toro Company
8111 Lyndale Ave. S
Bloomington, MN 55420
(612) 888-8801
Fax: (612) 887-8258

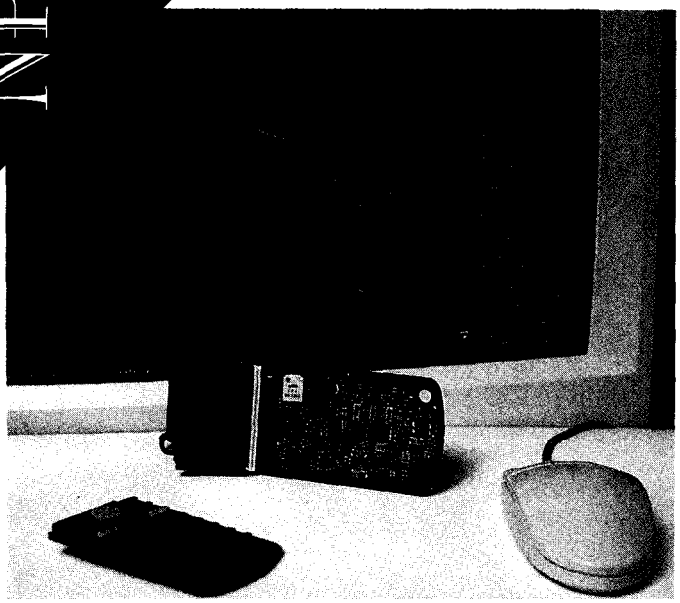
Orbit Underground Sprinklers
1065 N. 500th St.
Bountiful, UT 84010
(801) 299-5555
Fax: (801) 2995549

L.R. Nelson Corp.
7719 N. Pioneer Ln.
Peoria, IL 6 16 15
(309) 690-2200
Fax: (309) 692-5847

Rain Bird Co.
7590 Britannia Ct.
San Diego, CA 92173
(619) 661-4200
Fax: (619) 661-4283

I R S

407 Very Useful
408 Moderately Useful
409 Not Useful



HAND-HELD COMPUTER

The HHC386 is a new concept in ultracompact battery-powered hand-held computers designed for rugged industrial applications. Integrating a DOS core with unique peripherals such as laser diode bar-code scanners, vehicle sensors, industrial data processors, and Radio Frequency Identification (RFID) sensors, the system serves a variety of industrial needs. Typical uses include package tracking, factory data collection, and in-vehicle computers.

The system includes a 33-MHz AMD/'386 Elan CPU, serial port, parallel port, real-time clock, and other standard PC functions. Additional features include a NAND flash solid-state disk, 0.5–2 MB of RAM, 8- x 20character display with backlight and graphics, 41 -key keyboard, IrDA communications link, laser scanner interface, and an expansion interface for an RFID link.

A custom gate array contains hardware keyboard scan logic, random space-saving logic, and special data-acquisition hardware to process bar-code data. An intelligent charging system enables the unit to operate up to 100 h on a 9-V battery. A heater can be fitted to the unit to provide extended temperature control to meet severe outdoor requirements.

OEM agreements and reseller pricing is available.

Productivity Enhancement Products
26072 Merit Cir., Ste. 1 10
Laguna Hills, CA 92653
(714) 348-1011 • Fax: (714) 348-1310
http://www.pepinc.com/

#510

EMBEDDED CONTROLLER

A controller board optimized for use in embedded applications such as machine control or factoryequipment automation has been released by Arcom. The Target 188EB features a PC/104 expansion interface and an optional STE bus (IEEE-1 000), allowing economical expansion to match applications from simple machine controllers with PC-style peripherals to comprehensive rack-based solutions for advanced systems.

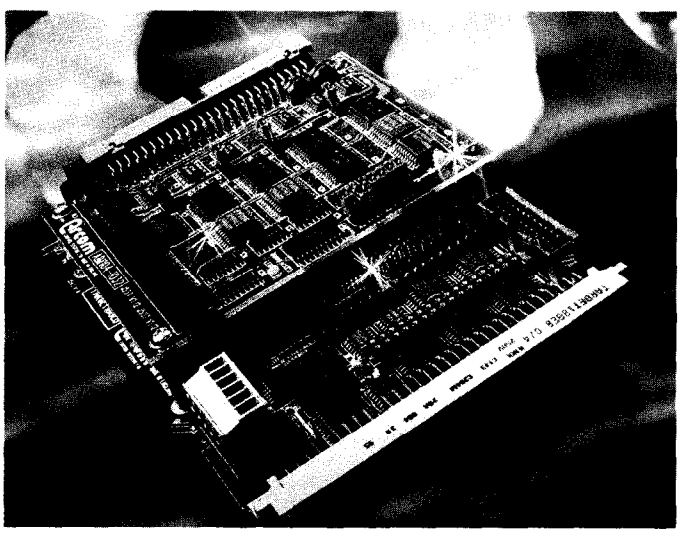
Based on the 25-MHz Intel 188EB embedded processor, this controller can run PC-compatible code and is designed to run without a BIOS or operating system. The application accesses all hardware directly. This approach to real-time system design delivers increased performance and saves significant system cost compared to a fully PC-compatible embedded controller.

Two versions of the Target 188EB are available. The standard version offers sockets for up to 0.5 Mb of flash or conventional EPROM and 0.25Mb SRAM, dual serial ports, 24 DIO lines, watchdog timer and power monitors, three counter timers, and an 8-bit PC/I 04 interface. It is geared for simple system needs such as an SBC or PC/104 stack. An alternative version offers an additional STE-bus expansion interface for applications requiring high I/O functionality and heavy-duty real-world interfaces. Both PC/104 and STE-bus interfaces may be used together for maximum flexibility.

The Target 188EB sells for $250 or, with an extra STE-bus interface, for $390. PC-hosted software development tools are also available as part of a development kit.

Arcom Control Systems, Inc.
135 10 S. Oak St. • Kansas City, MO 64145
(816) 941-7025 • Fax: (816) 941-0343

#511



Nouveau PC

edited by Harv Weiner

REAL-TIME MULTITASKING SYSTEM

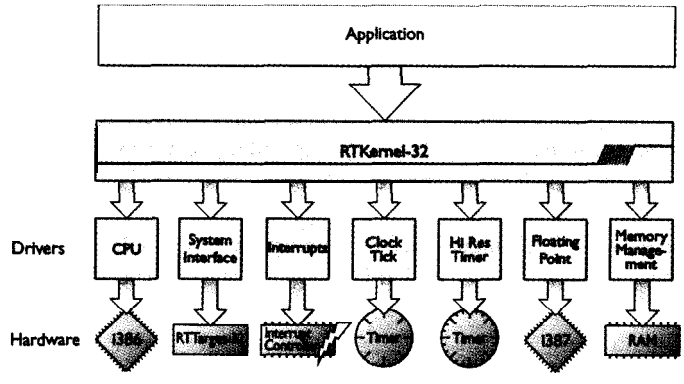
RTKernel-32 is a powerful real-time multitasking system for Intel-based 32-bit embedded systems and was specifically designed for 32-bit flat-address space environments. It is compact (about 16-KB code, 6-KB data) and provides the programmer with the basic tools needed to develop efficient real-time software.

RTKernel-32 is a library which can be linked to C/C++ application programs and offers functions to manage tasks, semaphores, mailboxes, interrupts, and so on. All RTKernel-32 tasks run within a single program. An application consists of an executable file with the kernel, required drivers, and all tasks. The executable can be processed by crossdevelopment systems for embedded systems supporting Intel '386 32-bit protected mode.

In addition to its native interface, RTKernel-32 offers an interfacecompatiblewith RTKernel-C, as well as Win32-compatible functions, making it possible to support the C/C++ compilers' Windows NT MultiThread Libraries. RTKernel-32's Win32 emulation covers about 30 Win32 API functions. Win32 Thread Variables are supported and duplicated for each task automatically at run time.

RTKernel-32 is portable, having been implemented completely in ANSI C. All CPU- and system-dependent parts of the kernel are encapsulated in replaceable device drivers. RTarget-32 is a rich set of drivers for use with the company's cross-development system. Third-party crossdevelopment systems can be used with the appropriately adapted drivers. RTKernel-32 supports Borland C/C++, Microsoft C/C++, and Watcom C/C++.

An RTKernel-32 developer's license is priced at $1950. Complete source code is available at an extra cost of $1650. RTTarget-32 sells for $1700, with complete source code costing an additional $1000. No run-time royalties are charged.



On Time Marketing
88 Christian Ave.
Setauket, NY 11733
(516) 689-6654
Fax: (5 16) **689-** 1172
info@on-time.com
http://www.on-time.com/

#512

CPU MODULE

The **CMV586DX133cpuModule** features the performance of a Pentium and the functionality of an embedded DOS controller on a PC/I 04 form factor. When stacked with appropriate video and data modules, it forms a complete embedded process-control system.

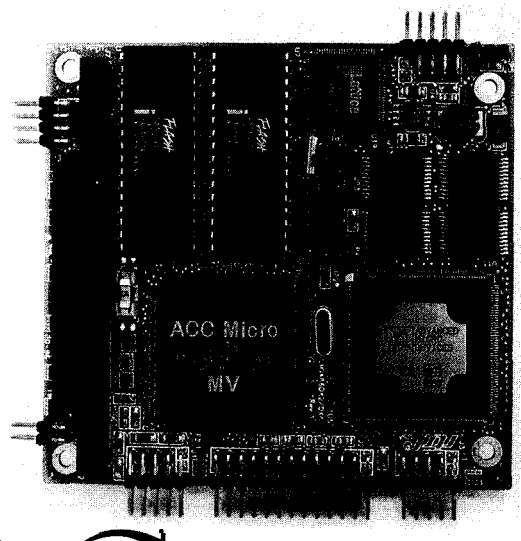
The module features the 133-MHz AMD Am5x86 processor that exceeds Pentium 75 performance with its 4x clock, internal math coprocessor, and 16-KB cache. It includes 8 MB of 32-bit DRAM, a PC/104 bus, two RS-232 serial ports, bidirectional parallel port, AT keyboard, speaker port, and a watchdog timer. The module also has flash file-system software for 2-MB EPROM, 1 -MB flash, or SRAM. M-Systems' DiskOnChip is supported as well.

The services and functions provided by the embedded BIOS ensure PC/AT compatibility. BIOS enhancements provide quick boot, virtual devices, and solid-state disk. Virtual devices let the operator use the keyboard, video, floppy, and hard disk on another PC-compatible computer through the serial port. A nonvolatile configuration EEPROM stores the system setup for battery-free operation. The module operates under MS-DOS, ROM-DOS, DR-DOS, Windows, OS/2, UNIX, QNX, RTXC, and other RTOSs.

The CMV586DX133cpuModule sells for $995.

Real Time Devices, Inc.
200 Innovation Blvd.
State College, PA 16804-0906
(8 14) 234-8087 • Fax: (8 14) 234-52 18
http://www.rtdusa.com/

#513



NPC

CROSS-PLATFORM DEVELOPMENT SYSTEM

The Willows Toolkit ports Windows applications to UNIX, Macintosh, and Java, as well as embedded systems. It also supports Microsoft Foundation Class (MFC) and Borland ObjectWindows Library (OWL) components.

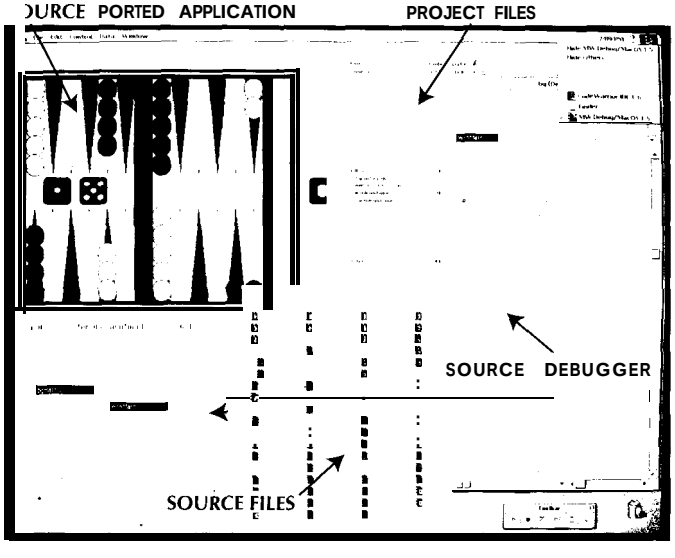
The toolkit has three major components-TWIN32 and -16 libraries, a platform abstraction layer, and a binary interface. Virtually identical libraries between platforms facilitate a single source tree. The platform layer contains platform-dependent code which takes advantage of the target platform's specific functionality and native performance. Library source code is included.

A Windows-compatible application interacts with the Willows library the same way the application works in a Windows environment. It calls API functions, receives messages, loads resources, and launches other Windows applications and DLLs. The toolkit also includes shell tools to convert sources and a resourcecompiler. Module-definition files create dynamically loadable objects to preserve the application's original architecture.

Binary emulation demonstrates how an application behaves before starting a port. The binary interface provides cross-platform native implementation of the Willows API. This unique facility intermixes source and binary modules, so applications can load resource files, bitmaps, and icons from existing DLLs without modification.

The Willows Toolkit for UNIX costs $4995. The Toolkit for Macintosh sells for $995.

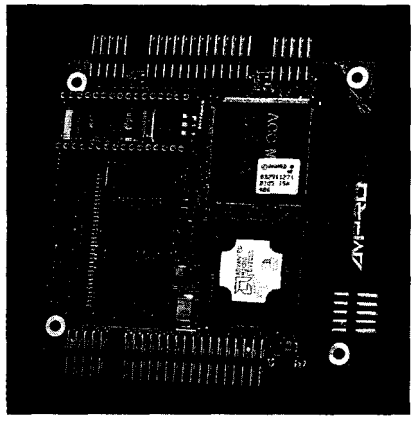
Willows Software, Inc.
12950 Saratoga Ave. • Saratoga, CA 95070
(408) 777-I 820 • Fax: (408) 777-I 827
http://www.willows.com/ **#514**



RUGGEDIZED PC/I 04 CPU

The **CoreModule/4DXi** is specifically targeted to hostile embedded environments requiring ruggedized products. It implements all the key functions of a full PC/AT-compatible system, including a '486DX4100-MHz CPU, 7 DMA channels, 14 interrupts, 3 programmable counter/timers, keyboard port, 2 serial ports, and floppy and fast IDE drive controllers. It has up to 52-MB DRAM, watchdog timer, real-time clock, and a bootable solid-state disk (SSD) option on a single PC/I 04 module.

To satisfy the demanding thermal requirements of mobile and portable embedded applications, the Small Quad Flat Pack (SQFP) '486DXIC package was used, along with a heatsink that fits within PC/I 04 space constraints. Extensive power-management support is included in both hordware and software, including BIOS functions compatible with the Advanced Power Management (APM) specification.



As a result, the module supports a standard operating temperature range of 0–70°C. An extended temperature rangeof-40" to +85°C is available by special order. To further accommodate tight power- or thermal-management requirements, a jumper option reduces the CPU clock rote from 100 to 66 MHz. Typical active operating power consumption is around 5 W at 100 MHz and 4 W at 66 MHz. In APM stand-by mode, this reduces to 600 mW.

The Module's floppy and IDE drive controllers load or store programs and data on standard disk drives. However, for enhanced reliability in harsh environments, a variety of SSD devices can be substituted. A 32-pin SSD socket allows an EPROM, nonvolatile RAM, or flash-memory chip to function as a bootable SSD drive. The SSD socket also accepts the high-density M-Systems DiskOnChip flash device, which offers up to 8 MB of read/write SSD. A high-capacity ATA-Flash SSD drive or card can also be cabled to the Module's IDE interface.

OEM pricing for the CoreModule/4DXi is $600 with 4-MB DRAM.

Ampro Computers, Inc.
990 **Almanor** Ave. • Sunnyvale, CA 94086
(408) 522-2100 • Fax: (408) 720-I 305
http://www.ampro.com/

#515

NouveauPC

NPC

ANALOG I/O MODULE

The Diamond MM- 16 PC/I 04 data-acquisition module provides a complete set of I/O features including 16 channels of analog input with 16-bit resolution, 4 channels of analog output with 12-bit resolution, and 16 lines of DIO. Analog inputs can be sampled at up to 100,000 samples per second using both internal and external triggering. Input range and polarity settings are programmable.

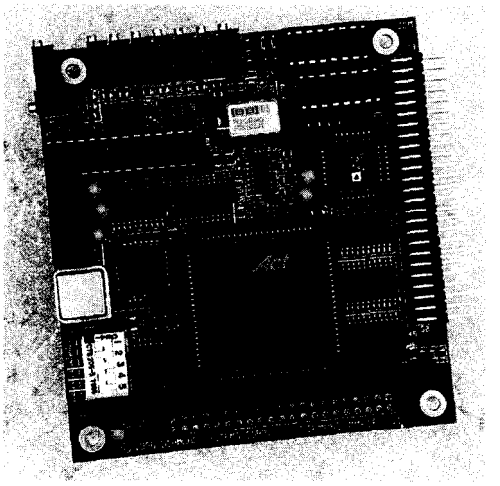
Key analog I/O settings are software controlled, including programmable gains of 1, 2, 4, and 8, as well as uni-/bipolar mode and analog-output uni-/bipolar mode. Digital output lines reset to zero to guarantee the board is in a predictable state on powerup or system reset.

The MM-I 6 requires only the basic +5-V system power supply for operation. The analog power is provided with an onboard miniature DC-to-DC converter that isolates the analog circuitry from digital system noise. Standard temperature operating range is from -25 to +85°C.

The Diamond MM-I 6 board is available in two versions. With analog outputs, the board is priced at $525. Without analog outputs, it costs $469.

Diamond Systems Corp.
450 San Antonio Rd. • Palo Alto, CA 94306
(415) 813-1100 • Fax: (415) 813-1130
http://www.diamondsys.com/

#516



NouveauPC

Mike Baylis

Get on the Bus!

The Evolution of the STD Bus to 32 Bits

Mike time travels though the past, present, and future of the STD bus. What *began as just a smaller,. more rugged form factor for industrial environments has grown to 32 bits with hot-swap capabilities. Where will it go from here?*

Let's face it. There are a lot of different bus architectures out there designed to address a lot of different application requirements-from small PC/I 04-bus form factors, through the ISA bus, and all the way up to high-end 6U VME systems.

The STD bus is flexible enough to offer solutions for application requirements ranging from embedded single boards to higher end multiprocessing systems.

If you're not familiar with the STD bus or if you have perhaps presumed it to be an extinct 8-bit platform, take a closer look at this versatile bus architecture. I'll start with a brief history of STD and move on to more technical discussions of STD80 and STD 32.

WHY ANOTHER BUS?

let's go back to the mid1970s. In an Intel Multibus I-dominated microcomputer market, a need existed for a smaller, rugged, and more cost-effective form factor for industrial environments.

With this in mind, the STD Bus Specification developed in the latter half of the '70s was quickly adopted by several manufacturers, most notably by Pro-Log and

Bus	Pin	Mnemonic	Signal Flow	Description	BUS	Pin	Mnemonic	Signal Flow	Description
Component Side					Circuit Side				
Logic	1	+5VDC		Logic Power	Logic	2	+5VDC		Logic Power
Power	3	GND		Logic Ground	Power	4	GND		Logic Ground
	5	VBATT	I	Battery Power		6	'DCPWRDWN	I	DC Power Down
Data	7	D3 (A19)	I/O (O)	Low Order	Data	8	D7	I/O (O)	High Order
	9	D2 (A18)	I/O (O)	Low Order		10	D6	I/O (O)	High Order
	11	D1 (A17)	I/O (O)	Low Order		12	D5	I/O (O)	High Order
	13	D0 (A16)	I/O (O)	Low Order		14	D4	I/O (O)	High Order
Address	15	A7	O	Low Order	Address	16	A15	O	High Order
	17	A6	O	Low Order		18	AI4	O	High Order
	19	A5	O	Low Order		20	AI3	O	High Order
	21	A4	O	Low Order		22	AI2	O	High Order
	23	A3	O	Low Order		24	AII	O	High Order
	25	A2	O	Low Order		26	A10	O	High Order
	27	A1	O	Low Order		28	A9	O	High Order
	29	A0	O	Low Order		30	A8	O	High Order
Control	31	*WR	O	Write Mem orI/O	Control	32	'RD	O	Read Mem orI/O
	33	*IORQ	O	I/O Address Select		34	*MEMRQ	O	Mem Address Select
	35	IOEXP	O	I/O Expansion		36	MEMEX	O	Mem Expansion
	37	*INTRQ1	I	Interrupt Request 1		38	'MCSYNC	O	CPU Mach Cycle Sync
	39	*STATUS1	O	CPU status 1		40	*STATUS0	0	CPU status 0
	41	*BUSAK	O	Bus Acknowledge		42	*BUSRQ	I	Bus Request
	43	*INTAK	O	Interrupt Acknowledge		44	*INTRQ	I	Interrupt Request
	45	*WAITRQ	I	Wail Request		46	*NMRQ	I	Nonmaskable Interrupt
	47	'SYSRESET	O	System Reset		48	'PBRESET	I	Push-button Reset
	49	'CLOCK	O	Processor Clock		50	*CNTRL	I/O	Aux Timing
	51	PC0	O	Priority Chain Out		52	PCI	I	Priority Chain In
Auxiliary	52	AUX GND	I	Aux Ground (bused)	Auxiliary	54	AUX GND	I	Aux Ground (bused)
Power	55	AUX +V	I	Aux Positive (+12 VDC)	Power	56	AUX –V	I	Aux Negative (-12 VDC)

Table 1: In looking at the ***STD-80*** bus *pin assignments, note that the address* lines ***A16–A19*** are multiplexed on data *lines DO-03 on each address cycle.*

Ziatech corporations. As a modular 8-bit data and 20-bit address bus with cards measuring only 4.5" x 6.5", STD bus **systems** quickly gained popularity in applications where issues such as harsh environments and size were critical.

Initial STD designs were based on Z80 and 6809 processors. However, thisquickly changed after the introduction of the 8088-based IBM PC in 1983.

That year, the 8088 processor became the basis for the STD-80 bus specification developed by the STD Bus Manufacturer's Group (STDMG). An Intel processor platform on the STD bus enabled engineers to use the PC softwaredevelopment tools and application software already in use on their desktop machines.

In fact, many PC advocates switched over to the STD bus to take advantage of its small, rugged nature. Today, several companies manufacture STD-80 and **STD 32** boards, and the number keeps growing [see sidebar "STD Bus Manufacturers").

Before delving into the technical details **of** both the STD-80 and STD 32 buses, I'll start with the STD-80 specification.

A CLOSER PEEK AT STD-80

As I mentioned, the mechanical form **factor for an STD-80 card is 4.5" x 6.5"** with a 56-pin card-edge connector. Electrically, these 56 bus pins are divided into four functional groups (i.e., buses)-data, address, control, and power.

The data bus is an 8-bit bidirectional **bus driven with tristate drivers. Data is** transferred onto STD pins **7-I 4** (data lines DO-D7). Read (*RD), write (*WR), and interrupt acknowledge (*INTAK) control signals affect data direction.

The address bus provides 20 address lines for memory **decoding and 16 lines for** I/O decoding. Memory request (*MEMRQ) and I/O request (*IORQ) control lines determine the operation.

This bus is also tristated, so it can be released to other bus masters. While address lines AO-A15 are driven onto STD pins 15-30, lines Al 6-A19 are multiplexed on data lines DO-D3 to enable addressing up to **1 MB of memory.**

Now, let's talk about some control-bus signal lines (I've already mentioned *RD, *WR, *MEMRQ, *IORQ, **and** *INTAK). **Another address qualifier control line, IOEXP, enables** 8-bit **I/O port mapping for older** 8-bit **STD I/O cards.**

***MCSYNC on STD pin 38 is a synchronization signal asserted to indicate a bus transaction when** *RD, ***WR, or** *INTAK **is asserted. This pin can also be used for Address Latch Enable.**

***BUSRQ and** *BUSAK **are control lines used by bus masters to request and relinquish bus control. For interrupt support,** *NMIRQ **on pin 46 is for nonmaskable interrupts, and** *INTRQ **and** *INTRQ1 **are generic** maskable **processor-card interrupts.**

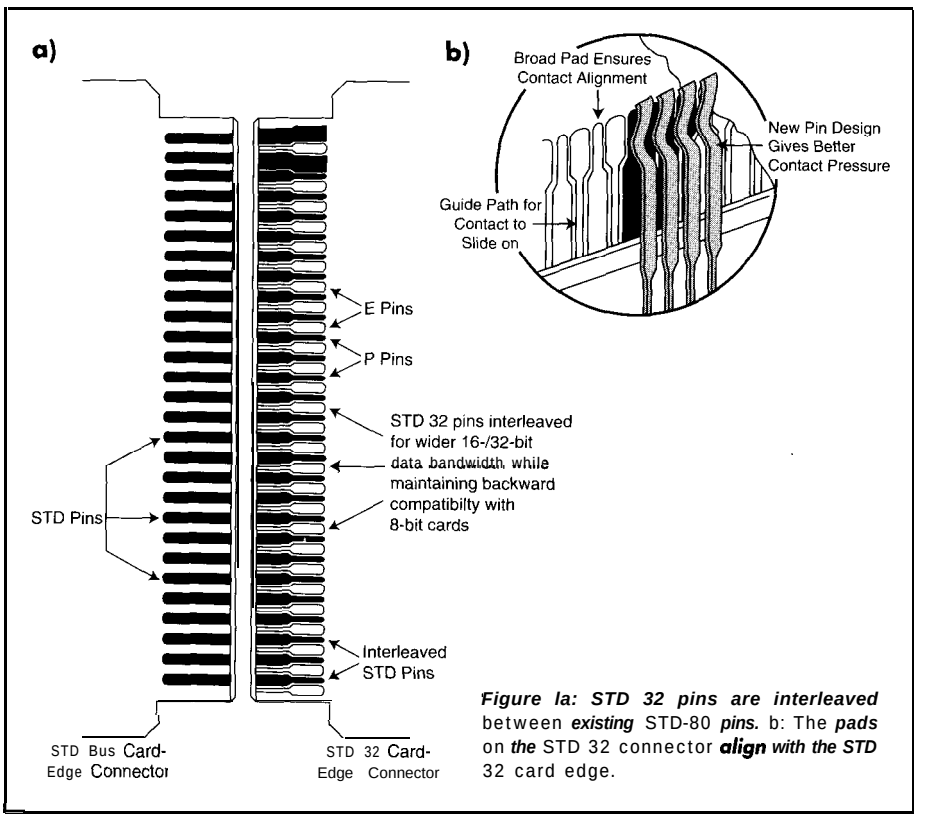


Figure la: STD 32 pins are interleaved between ***existing*** STD-80 ***pins.*** b: The ***pads*** on ***the*** STD 32 connector ***align with the STD*** 32 card edge.

Pin	Mnemonic	Signal Row	Description
Component Side			
1	GND	I	Logic Ground
3	XA19	O	Address
5	XA18	O	Address
7	XA17	0	Address
9	XA16	0	Address
11	*NOWS	I	No Wait States
13	+5VDC		Logic Power
15	*DAKx	0	DMA Acknowledge
17	GND	I	Logic Ground
19	D27	I/O	Data
21	D26	I/O	Data
23	D25	I/O	Data
25	D24	I/O	Data
27	D23	I/O	Data
29	D22	I/O	Data
31	D21	I/O	Data
35	D20	I/O	Data
37	GND	I	Logic Ground
	D19	I/O	Data
39	D18	I/O	Data
41	D17	I/O	Data
43	D16	I/O	Data
45	GND	I	Logic Ground
47	IRQx	I	Interrupt Request
49	*BE1	O	Byte Enable 1
51	*BE0	O	Byte Enable 0
53	*MEM16	I	Memory 16-bit
55	M-IO	0	Memory or I/O
57	*DMAIOW	0	DMA I/O Write
59	*IO16	I	I/O 16-bit
61	'CMD	0	Command
63	*EX16	I	Exchange 16-bit
65	EXRDY	I	Exchange Ready
67	*INTRQ3	I	Interrupt Request 3
69	*MAKx	O	Master Acknowledge
71	'SLBURST	I	Slave Burst
73	*XA27	O	Address
75	*XA26	O	Address
77	'XA25	O	Address
79	*XA24	0	Address

Pin	Mnemonic	Signal Row	Description
Circuit Side			
4	'LOCK	O	Lock
6	XA23	O	Address
	XA22	O	Address
10	XA21	O	Address
12	XA20	0	Address
	RSVD	–	Reserved
14	+5VDC	I	Logic Power
16	*DREQx	I	DMA Request
18	GND	I	Logic Ground
20	D31	I/O	Data
22	D30	I/O	Data
24	D29	I/O	Data
26	D28	I/O	Data
28	GND	I	Logic Ground
30	D15	I/O	Data
32	D14	I/O	Data
34	D13	I/O	Data
36	D12	I/O	Data
38	D11	I/O	Data
40	D10	I/O	Data
42	D9	I/O	Data
	D8	I/O	Data
48	*MASTER16	O	Master 16-bit
46	*AENx	O	Address Enable
50	*BE3	0	Byte Enable 3
52	'BE2	0	Byte Enable 2
54	GND	I	Logic Ground
56	W-R	O	Write or Read
56	'DMAIOR	0	DMA I/O Read
60	*EX8	I	Exchange a-bit
62	'START	O	Start
64	*EX32	I	Exchange 32-bit
66	T-C	I/O	Terminate or Count
66	+5VDC	I	Logic Power
70	*MREQx	I	Master Request
72	*MSBURST	O	Master Burst
74	*XA31	O	Address
76	*XA30	O	Address
78	*XA29	O	Address
80	*XA28	0	Address

Table 2: ***STD*** 32 ***E*** pins and signals are used to expand the ***STD-80*** subset. Original STD-80 signals are ***implemented on P pins on the STD 32 connector.***

Clock and reset control signals include *CLOCK for system synchronization, *SYSRESET, which is triggered by power-on detection or a push-button reset, and *PBRESET for a debounced push-button reset. Both control lines reset the processor when asserted.

And, don't forget the power bus. It handles logic (+5 V) and auxiliary (±12 V) power distribution. You'll find a complete STD80 bus pin description in Table 1.

A BIT CONSTRAINED

As Intel processors progressed from 8- to 16- and 32-bit architectures, it seemed obvious that the STD-80 bus's 8-bit data path would become a bottleneck. Some initially argued that 16- and 32-bit transfers were only needed between the CPU and onboard peripherals (e.g., memory).

Soon, though, major STD vendors developed a 16-bit extension to the STD-80 bus. Referred to as P16, this extension multiplexed the additional data and address lines over existing lines. Unfortunately, it has its own problems and inconsistencies.

First, multiplexing data on address lines A8–A15, which STD-80 defines as valid through a whole data cycle, can cause an STD-80 card to decode this data as an address and respond. This possibility severely limits usable address space.

Also, not all STD manufacturers used the same backplane timings and byte laning (i.e., 8-bit odd and even bus transfer).

AN INNOVATIVE SOLUTION

To address these limitations and inconsistencies, Ziatech introduced STD 32 in 1989 as an extension to the STD-80 bus. Its philosophy is simple-support higher data transfers and true multiprocessing, while maintaining backwards compatibility.

How is this accomplished? Incorporated EISA-like signals add capabilities such as extra data and address lines for 16- and 32-bit systems, bus arbitration for multiprocessing support, additional interrupt lines, and DMA capability.

And, a new connector scheme that interleaves new STD 32 signals with existing STD80 signals (as shown in Figure 1) lets STD80 cards be used in STD 32 backplanes. This connector is also more reliable, offering better contact pressure, conductivity, and shock/vibration resistance than the connector used on STD-80 designs.

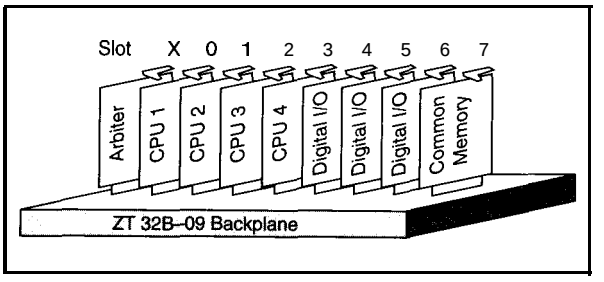


Figure 2: When you look at the bus *layout* of a STD 32 multiprocessing system, note the *arbiter card located in slot X and the shared memory card* used for interprocessor ***communication.***

Let's get into a more detailed technical discussion about STD 32 and the possibilities it opens up.

STD 32

STD 32 supports 8-, 16-, and 32-bit data transfers. It defines five data-transfer types between bus masters and peripheral boards--8and 16-bit standard architecture (SA) and 8-, 16-, and 32-bit extended architecture (EA).

The 8-bit standard-architecture (SA8) transfers are the default STD 32 data-transfer mode and are de signed to be compatible with STD-80 cards. If the peripheral board drives STD 32 signal *EX8, *EX16, or * EX32 at the start of this transfer, then an EA transfer occurs.

The nominal clock cycle for an SA transfer is five *CLOCK cycles, and for EA transfers, two *CLOCK cycles. Burst EA transfers in which data transfers every ***CLOCK** cycle are possible if the peripheral card drives STD 32 signal *SLBURST while the bus master is driving STD 32 signal *MSBURST. Also, for increased throughput, EA data transfers don't multiplex data and address lines.

A total of five backplane interrupt lines are now supported with the addition of STD 32 signals *INTRQ2–*INTRQ4. A list of the additional STD 32 signals is summarized in Table 2.

Also, each of the first 15 slots of the STD 32 backplane has a dedicated interrupt routed to Slot X, the leftmost STD 32 connector. This slot is important **for** multiprocessing sup port-one of the great benefits of STD 32.

STD 32 implements multiprocessing through an architecture called Multiple Instruction, Multiple Data (MIMD) with distributed memory.

Essentially, multiple processors perform instructions on their own in-memory data with interprocessor communication managed through a common memory area. Bus contention is managed through a centralized arbitration scheme.

Master request (*MREQx) and Master Acknowledge (*MAKx) signals are routed from the first seven STD 32 slots through the backplane to Slot X. Each multiprocessing CPU accesses the backplane to read or write to a memory or I/O card.

When this happens, the CPU asserts the *MREQx signal, which routes to an arbiter card in Slot X. This arbiter card either grants or denies access to the bus depending on whether the bus is free or another multiprocessing CPU is accessing the bus.

If it's free, the arbiter card asserts *MAKx. In it's busy, the arbiter card waits until the current bus cycle is complete and grants access based on a rotating priority scheme. This system ensures that each CPU gets equal access to the STD 32 bus.

The interprocessorcommunication component is handled through a backplane memory card addressable by any CPU. Finally, CPUs use a common interrupt line to notify each other of pending messages or a request for attention. Figure 2 shows the STD 32 multiprocessing approach.

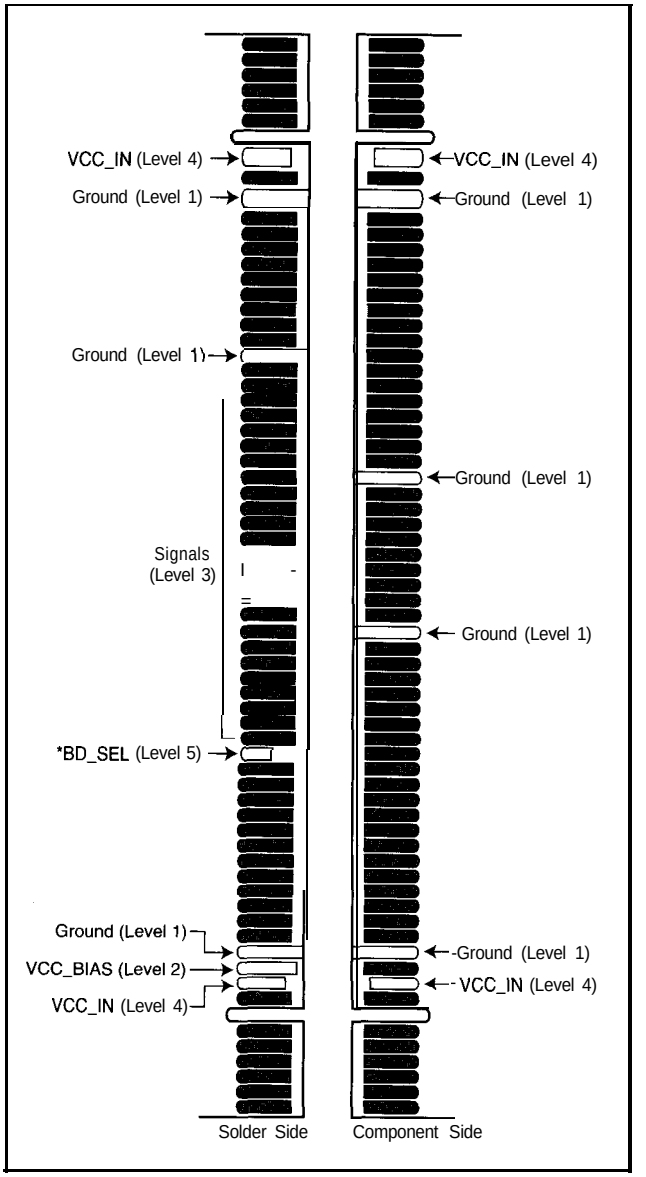


Figure 3: The staggered finger pattern implements hot swap on STD 32 boards. *This* pattern sequences the power to *prevent backplane signal g/itching.*

THE REAL VIEW

How **about a** practical example of using STD bus?

Suppose you have an application requirement for real-time control in a factory environment, yet you also need to support a Windows-based graphical user interface. The Windows GUI requires a PC-compatible architecture, but Windows can't satisfy real-time needs.

It's best to take a multiprocessing approach to this problem. A '486 or Pentium-based multiprocessing CPU can satisfy the GUI requirements. And, real-time control can be managed by a second or even third processor.

Since disk resources can be shared in a STD 32 multiprocessing environment, both CPUs can load required software from a hard disk. The real-time CPU can pass data to the Windows CPU through the shared memory card.

Several software methods are available for exchanging this information. The real-time CPU can write to shared memory directly by declaring a far pointer in C or using Peek and Poke calls in BASIC to the physical address where

Level	Function	Stagger	Distance	Description
1	Ground	0.000"		Ground is the first level connected to the backplane.
2	VCC_BIAS	0.020"		VCC_BIAS provides power to the control logic that keeps the backplane buffers disabled on powerup. Logic using VCC_BIAS should be kept to a minimum to avoid glitching the power supply.
3	Signals	0.060"		Backplane interface signals
4	VCC_IN	0.080"		VCC_IN to the board that is inrush limited
5	*BD_SEL	0.160"		Board select enables power-up circuitry onboard. The backplane interface logic is tristate until this pin is connected and onboard VCC reaches 4.75 V.

Table** 3: As a hot-swap **card is inserted or** removed, the changes in distance between the staggered fingers ensure that the **signals connect or disconnect in the proper sequence.

this memory resides. Or, you can use an NDIS driver to make the backplane and shared memorycardappearasan Ethernet card to each of the processors.

On the Windows CPU side, a DDE (Dynamic Data Exchange) driver can be used that takes data out of shared memory and makes it available to Windows applications as DDE variables. Or, again, you could use the NDIS driver.

The arbiter card in Slot X prevents the CPUs from simultaneously **accessing** shared memory. Even though the arbiter card prevents a collision between CPUs, shared data can become corrupted if precautions are not taken.

Suppose CPU 1 changes a semaphore in shared memory after CPU0 has read the semaphore but before modifying and writing it back. CPU I's write will be lost.

There are two solutions to this problem. The Lock signal from a CPU can be implemented to preserve a memory location during an update, or you can manage it with Ziatech Star BIOS function calls.

FUTURE DIRECTIONS

Many industries such as telecommunications, medical, and manufacturing may require a fault-tolerant computer system (i.e., a system that can continue operation after a hardware or software fault).

To better support such critical applications, Ziatech introduced an extension to the STD 32 specification to support hot-swap capability in STD 32 systems. Hot-swap capability lets you replace a faulted board without interrupting system operation.

Several issues had to be addressed to support hot swap. for one, CMOS, ACL, and ACT logic families have no isolation when power is removed. They contain P-channel **transistors** or diodes between the signal pin and VCC that become forward biased when VCC is removed.

This change disturbs the bus by pulling it one diode drop above ground. ABT-type parts from National Semiconductor do not have clamping transistors or diodes, and they guarantee input and output will be off when VCC is applied or removed.

Second, ABT parts buffer-output enables must remain above rising and falling VCC levels until VCC stabilizes at 4.75 V, so power sequencing is required to meet these criteria. A staggered card-edge finger pattern, as in Figure 3, provides the necessary power sequencing. Table 3 summarizes the staggered finger-pattern levelsand their functions.

Also, when inserting or removing a hot-swap board, backplane voltages can be forced out of regulation unless power-control circuitry, limits the power-on/-off inrush current.

Keep in mind that this STD 32 bus extension and the hot-swap cards only address the hardware aspects of a fault-tolerant system. Application software must know when an I/O board is removed or inserted and manage the switchover to this new board. So, careful software design is crucial for these systems.

Supporting this software aspect, a powercontrol register resides on the hot-swap board. Software can read and write bits in this register to enable or 'disable power to a board or determine whether a board was removed and reinserted.

HEADING HOME

I've shown you just how versatile and well-suited this bus is to applications that have a PC platform in a small, rugged form factor. It has enough flexibility to provide a solid upgrade path should higher performance be needed in the future.

Next month, I'll explore a higher bandwidth embedded-PC architecture that's attracting a lot of attention. EPC

Mike *Baylis* is an *applications* engineer *for* Ziatech, a *manufacturer of small embedded* computers for *STD* 32 and *CompactPCI* Bus. You may *reach him at* mike_*baylis@ziatech.com*.

REFERENCES

A. Beverly. "Fault-Tolerance Systems: Highly Available **on** Industrial Computer Standards." *RTC* **4:6, 1996.**

A. Beverly. "Hot-Swap and Fault Tolerance on STD 32." App Note, 1995.

STD Manufacturers Group. *STD-80* **Series** *Bus Specification and* **Designers Guide, Rev. 2.3, 1989.**

STD Manufacturers Group. SJD **32** *Bus* **Specification** *and Designers Guide,* **V. 1.3,** *1994.*

SOURCES

STD-80 and STD 32 boards, Hot-swap board, ***STD*** 32 ***Bus Specification and Designers Guide,*** and "Hot-Swap and Fault Tolerance on STD 32."

Ziatech Corp.
1050 Southwood Dr.
San Luis Obispo, CA 93401
(805) 541.0488
Fax: (805) 541.5088
BBS: (805) 541-82 **18**

ABT-type parts
National Semiconductor
2900 Semiconductor Dr.
Santa Clara, CA 95052
(408) 72 I-5000

STD Bus Manufacturers

Analog Devices, Inmation
APIX, Inc.
Bus Technologies
Carroll Touch System Products
Communication Automation & Control, Inc.
Computer Dynamics
Contemporary Control Systems, Inc.
Cytec Corp.
Dataforth Corp.
DataLink Technologies, Inc.
Datum Inc., Bancomm Timing Div.
DDI, Inc.
Digitron Electronics Co., Inc.
Elma Electronic Inc.
Engenuity Systems
Galil **Motion Control, Inc.**
GE Fanuc Automation North America, Inc.
General Digital Corp.
GreenSpring **Computers, Inc.**
Hybricon **Corp.**
IOTech, **Inc.**
Matrix Corp.
Micro-Aide, Inc.
Micro-Link **Technologies, Inc.**
Micro/sys, **Inc.**
Microcomputer Systems, Inc.
Mitchell Electronics
Motion Engineering, Inc.
National Instruments Corp.
Oregon Micro Systems, Inc.
Pro-Log Corp.
Robotrol **Corp.**
Sensoray **Co., Inc.**
Systek
Technology 80, Inc.
Twin Industries, Inc.
Universal Systems, Inc
Vector Electronic Co.
VersaLogic **Corp.**
Western Reserve Controls, Inc.
Whedco, Inc.
WinSystems, Inc.
XYZ Electronics, **Inc.**
Ziatech Corp.
Zwick **Systems**

IRS

410 Very Useful
41 1 Moderately Useful
412 Not Useful

Bill Payne

A Networking Primer

Part 2: LAN Media and Various Protocols

Tangled in too many networking problems caused by the incorrect use of *cable* types? *If* so, Bill is ready to straighten out the kinks. He explains mode*l*s, protocols, and implementations of cable specifications.

Last month, I presented the basics of a LAN environment, including the workings of a DOS redirector and IAN printing.

This month, I cover LAN media and protocols, which contain some of the most complex information and concepts of LANs and their operation. Impedance, propagation delays, and stub length are as important to a LAN as they are to any transmission system.

I discuss the specifics of cabling used to build a IAN and go over the basic OSI model and how it relates to existing protocols. After that, I review various common protocols, as well as some basic rules that ensure the proper operation of a LAN.

LAN MEDIA

Cabling is the most important part of any IAN. It contributes to over 80% of the problems associated with LANs.

Many people assume cable is nothing more than a pair of wires connecting multiple workstations. While this is basically true, it's like saying that any antenna, regardless of its length or impedance, works with any frequency transmitter. Both the antenna and transmitter must match to achieve optimum operation.

Similarly, IAN cabling must be matched to the needs of the system to operate at peak efficiency.

It seems there are as many standards in the LAN cabling industry as there are resistor values. Some even use the same designations and are easily confused. I've spent hours troubleshooting a problem only to find that someone inadvertently used the wrong type of cable.

With this in mind, I'll jump right into the IBM cabling standards and the EIA/TIA-568 standards.

The IBM cabling standards define nine cable types. While IBM created these specifications to provide a stable and known environmentfortheirequipment, theydon't manufacture the connectors or cable.

Although Table 1 lists the cable types, their composition, and specific uses, here are a couple of extra pieces of information.

Type 3 cable has a characteristic impedance of 105 Ω and is called Unshielded Twisted Pair (UTP). It's easily confused with the EIA/TIA Category 3 cable.

I've found many a Token-Ring LAN wired with STP cable and using UTP cable for patch cords. The problem is that STP cable has a characteristic impedance of 150 Ω. This impedance mismatch may seem negligible at first, but it causes problems with the reflection coefficients of the connection.

Type 5 cable uses a 100-µm fiber-optic core and measures 140 µm with cladding, which differs considerably from the more popular 62.5-µm fiber-optic core that measures 125 µm with cladding.

The only cable types not listed in Table 1 are Types 4 and 7, which lack a published specification from IBM.

The EIA/TIA-568 standard, created jointly by the Electronic Industries Association and the Telecommunications Industry Association, defines an open standard which doesn't carry the approval of any single vendor. It describes both the perfor-

mance characteristics of the cable and proper installation methods.

This standard specifies that data and voice cables are to be run to each network connection wallplate. As you see in Table 2, categories are used instead of types.

PROTOCOL MODELS

The remaining 20% of LAN problems involve protocol problems. They may be as simple as trying to connect a device using the IEEE 802.2 protocol on a IAN with IEEE 802.3 protocol. Let's look at the universal structure for developing and implementing protocols.

First, figure out whether it is an ISO or OSI model. Why the International Organization for Standardization (ISO) decided to name their model the Open Systems Interconnect (OSI) model is beyond me.

Now that you know the difference, I can explain what a model, protocol, and implementation are. A model represents the general concepts and guidelines of how to move data between network nodes. It details specific services offered and which layer is responsible for them.

The term "protocol" defines the set of rules concerning hardware, procedures, and data structures.

Implementation defines the way in which a developer creates a product based on the protocol specifications.

The seven-layer OSI model shown in Figure 1 is a logical framework for developing network protocols. Each layer is assigned a specific function.

The services layers (5, 6, and 7) are responsible for providing network services to user application programs.

The communications layers (3 and 4) ensure that data is transported reliably from network node to network node independent of the physical medium.

The physical layers (1 and 2) physically move the data over the network media. Let me review these definitions a bit more.

The Application Layer provides network services to the user and application programs. Application Layer protocols must not be confused with user applications. They are network service protocols accessed through Application Programming Interfaces (APIs).

The Presentation Layer prepares messages for the Application Layer, including file format conversion (e.g., EBCDIC to ASCII) and syntax translation between the hosts. Services such as data compression and encryption are also provided by it.

The Session Layer establishes and terminates communication sessions between hosts. It also negotiates connections, authenticates access, synchronizes dialogue, and providesoverall session management.

The Transport Layer maintains data integrity from source to destination via the communications medium. It ensures reliable communications between processes from end to end on the LAN.

The Network Layer delivers data packets across a network based on software addresses. It establishes a path from the source node to the destination node using these addresses. This function, referred to as routing, will be covered in more detail in Part 3 (INK 81).

The Data Link Layer builds Media Access Control (MAC) frames from the raw data bits and transports them between network nodes. It is the transitional interface between the network hardware and the upper layers of the model. While it's similar to the Transport Layer, it can only transfer data between adjacent nodes.

Finally, the Physical Layer describes the actual physical specifications of the communications medium. Specifications such as the type of cable, the electrical properties of the transmit and receive signals, and the procedural properties for accessing the medium are defined here.

Let's look into some of the more popular implementations of these protocols.

Type	Composition	Notes
1	2 twisted-pair 22-AWG solid wire shielded by external braid (i.e., STP)	150-R impedance, tested to 100-MHz bandwidth, Type 1 A-300 MHz, supports up to 1 00-Mbps comm system, common in Token-Ring installations
2	multiple unshielded 22-AWG solid wire and 2 shielded pairs meeting Type-I spec in same sheath	provides voice (unshielded) and data (Type 1) in same cable
3	4 unshielded-pair 24-AWG solid wire (i.e., UTP)	105-Ω impedance, easily confused with EIA/TIA Category 3 cable
5	2 fiber-optic strands	100-µm core, 140 µm with cladding, window at 850 nm, 100-MHz bandwidth
6	2 twisted-pair 26-AWG stranded wire in shielded braid	used as interface cable between Token-Ring network card and connector on wall plate
8	2 parallel-pair 23-AWG solid wire	used under flooring because cable has no twists
9	2 individually shielded twisted-pair 26-AWG solid or stranded wire	plenum cable, has fire-retardant coating, required between floors in a building and in overhead areas

Table 1: IBM specified these cable *types.* Types *1, 3,* and6 are the most common ***and are frequently found*** in Token-Ring environments.

Category	Composition	Notes
1	22- or 24-AWG untwisted wire	Not generally recommended, comes in a variety of attenuation and impedance values
2	22- or 24-AWG solid wire in twisted pairs	Typical in IBM 3270 and AS/400 environments, derived from IBM Type 3 cable specification
3	multiple twisted-pair 24-AWG solid wire	Lowest category for 10BaseT Ethernet installation, 100-Ω characteristic impedance, tested for attenuation and cross-talk through 16 MHz
4	22- or 24-AWG solid wire in twisted pairs	Found in installations running Token Ring at 16 Mbps, 100-Ω characteristic impedance
5	multiple unshielded twisted pairs of 22- or 24-AWG stranded wire	Recommended for all new installations, 100-Q characteristic impedance, tested for attenuation and cross-talk through 100 MHz

Table 2: Here are the cable types according to the ***EIA/TIA*** numbering *convention.* If you *compare Category* 3 cable with Type 3 cable, you can readily see why confusion abounds.

ETHERNET

Ethernet is based on a broadcast architecture in which every node receives every broadcast at approximately the same time. Itcommunicates using Manchester-encoded digital baseband signaling.

Ethernet protocol can be transmitted using coaxial, twisted-pair, or fiber-optic cable. It uses a Media Access Protocol referred to as Carrier Sense Multiple Access with Collision Detection (CSMA/CD).

Ethernet adapters listen before transmitting to avoid collisions of transmission media. Collisions can be detected on wire media by the higher electrical signal produced by simultaneous transmissions.

One drawback to Ethernet is its relatively low usable bandwidth. If many devices' average use of the bandwidth causes excessive collisions, a low throughput results.

Ideal utilization on a Ethernet LAN using coaxial cable is 1–25%. Marginal utilization occurs within 26-44%. Anything over 45% utilization is prohibitive (see Figure 2):

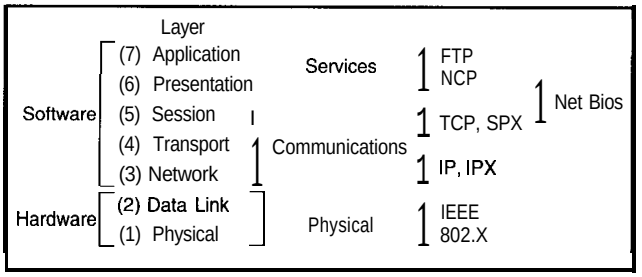


*Figure 1: The **OSI** model has seven distinct layers, each with a specific set of functions. Here, you see how the various protocol implementations are used in the model.*

- a trunk segment may not exceed 185 m
- a maximum of five trunk segments may be connected through four repeaters
- onlythreeof these segments may contain nodes. The other two are for distance only.
- theentire network may notexceed 952 m
- the minimum cable distance between nodes is 0.5 m
- the maximum number of nodes per trunk is 30 including the repeaters
- only one end of the cabling is tied to ground

Today, Ethernet is more commonly installed using twisted-pair Category 5 cabling. This is referred to as 1 0BaseT where 10 is the signaling speed in Mbps, Base signifies baseband signaling, and T designates twisted-pair cabling.

It is implemented in a star topology, where each node connects to a device (i.e., a hub) using RJ-45 connectors. Its rules are:

- each segment is limited to a maximum of 100 m
- the maximum number of trunk segments is 1024
- the 5-4-3 rule still applies (i.e., a maximum of five segments using four repeaters can exist. Only three segments can have nodes on them. Two are for distance only.)

TOKEN RING

Token Ring is a protocol which uses a special bit pattern, referred to as a token, to grant access to the ring. As its name implies, this protocol is based on a ring topology.

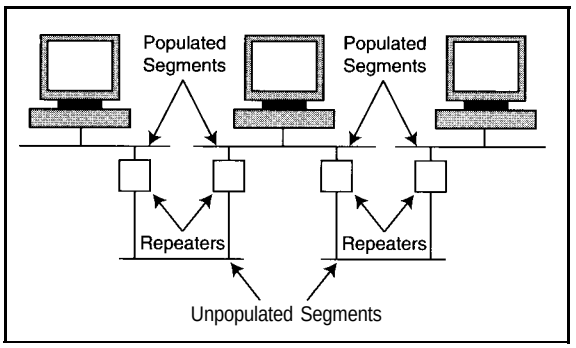


*Figure 2: The **Ethernet** S-4-3 rule states that there can be no more than five physical segments in the LAN, there can be a maximum of four repeaters, and **only three** of **the** five segments can be populated with nodes.*

The continuously rotating token provides a fault-tolerant deterministic access medium at an average to high cost. It's deterministic in that every node on it receives a free token within a specific period of time. Therefore, it's ideal for applications where delays are critical (e.g., robotics and process controls).

Signaling is accomplished by differential Manchester encoding. When a station has data to send, it changes the free token to a busy token and appends its message to the end. Each message is formatted in an application program into a packagecalled a frame before transmission.

Each node is attached to a device referred to as a Multistation Access Unit (MAU). The ring can operate at a signaling speed of either 4 or 16 Mbps.

The node with the highest internal address acts as the monitor to keep the token rotating. Unlike Ethernet, Token Ring's usable bandwidth is due to hardware limitations and not protocol.

An ideal utilization is from 1 to 65%. Marginal utilization occurswithin 66-79%. Utilization over 80% is prohibitive.

The rules for implementing Token Ring directly relate to the type of cabling used in an installation. IBM STP cabling can sup port up to 260 nodes on a single ring. IBM UTP cabling supports a maximum of 72 nodes per ring.

Token-Ring network interfacecards have extensive built-in media-management capabilities. Therefore, it's more expensive than all other media types except FDDI.

FDDI

Fiber Distributed Data Interface (FDDI) was the first access standard for fiber-optic media (ANSI X3T9.5). Like Token Ring, it rotates tokens on counter-rotating rings. This dual-ring architecture provides for the redundancy and fault tolerance of FDDI.

Due to the high cost of media and network interface cards, this technology is usually found only in LAN backbones. Its primary purpose is to tie multiple servers or other high-bandwidth devices together.

FDDI's three greatest attributes are bandwidth, distance, and its inherent immunity to electromagnetic interference (EMI). The ANSI X3T9.5 specification states a current bandwidth of 100 Mbps with a usable bandwidth of over 80%.

A T M

Asynchronous Transfer Mode (ATM) is part of the Broadband ISDN (BISDN) standard. ATM provides a unified communications infrastructure capable of supporting data, video, and voice.

ATM is based on a technology known as cell switching, which uses 53-byte packets, each referred to as a cell.

One of ATM's best features is its controlled communications latency. It becomes critical when a network must provide full-motion video with voice synchronization. ATM accomplishes this via small cell size, cell buffering, and high-speed lines.

ATM is scalable over physical network size, communications speed, and node count. It eliminates the distinction between LANs and Wide Area Networks (WANs).

The technology uses cell-signaling procedures toestablishvirtual circuits between nodes. A network layer referred to as the ATM Adapter Layer (AAL) performs the segmentation of raw data bits into cells and reassembly of cells into raw data.

There are five AAL types in the ATM specification. The AAL type used by the connection, along with the traffic type and speed, is determined at call setup.

There are three data-pacing mechanisms in the ATM environment. Constant Bit Rate (CBR) only sets the virtual circuit's peak transmission rate. Variable Bit Rate (VBR) sets the peak and sustained transmission rates. Available Bit Rate (ABR) sets the peak and minimum transmission rates.

The interface between the ATM switches is referred to as a Network-to-Network Interface (NNI). The interface between the ATM switch and the user is referred to as a User-to-Network Interface (UNI). Applications like multimedia drive this technology.

ARCnet

Datapoint developed ARCnet technology and was supported by Standard Microsystems. ARC stands for Datapoint's Attached Resource Computing architecture.

From a network standpoint, ARCnet is a verywell-behavedcommunicationsmethod. It does not use a collision-based protocol.

Instead, it uses a broadcast architecture in which all stations receive messages at about the same time. One station at a time receives the transmit permission token.

The installer must configure each network adapter with a node address between 1 and 255. When the network is first activated, each network adapter broadcasts its address on the LAN.

The node with the lowest address becomes the token controller for the LAN. After the controller is chosen, it begins a round-robin polling sequenceofeach node on the network. If a node has information to transmit, it must wait until it receives a permission token from the controller.

When a new node enters the network, all stations go into recon mode. During this reconfiguration, all nodes rebroadcast their address and a new controller is chosen.

ARCnet has traditionally been installed using RG-62U, 93-Ω coax cable using both passive and active hubs. The passive hubs function as signal splitters, while the active hubs function as amplifiers.

One of the features of ARCnet is its total network length. It cannot exceed 20,000 [i.e., propagationdelaymaximumof3 1 µs). The rules for ARCnet are:

- unused nodes on all passive hubs must be terminated using 93-Ω terminators
- a passive hub cannot connect to another passive hub
- passive hubs can only connect a node to an active hub, never two active hubs
- node-to-node distances cannot exceed 2000
- node-to-active-hub distances cannot exceed 2000
- node-to-passive-hub distances cannot exceed 100'
- active-hubtoclctive-hub distancescannot exceed 2000'
- active-hub-to-passive-hub distances cannot exceed *100*

CONCLUSION

Confusion abounds on the two primary cabling systems in use in industry. The OSI model shows how each layer functions and the differences between models, protocols, and implementations.

To go deeper into the framing formats for each protocol, check the contacts I've listed. In Part 3, I'll cover the devices used in an internetworking environment. EPC

Sill *Payne has many years' experience as a digital design engineer. He holds two semiconductorpatentsand has others pending. You may reach Bill at bpayne @kramerkent.com.*

CONTACTS

ARCnet
Datapoint Corp.
8400 Dotapoint Dr.
San Antonio, TX 78229-8500
(5 12) 593.7263
Fax: (5 12) 293-7355

Standard Microsystems Corp.
80 Arky Dr.
Hauppauge, NY 11788
(5 16) 4354000
Fax: (5 16) 435-6000

EIA/TIA-568 Standard
Electronic Industries Assn.
15 Inverness Way E
Englewood, CO 80112
(303) 290-9 159

IBM cabling standards
IBM Corp.
1000 NW 51 St.
Boca Raton, FL 33432
(407) 443-2000
Fox: (407) 443.4533

DEC
77 Reed Rd.
Hudson, MA 01749
(508) 568-6868
Fox: (508) 568-6447

Novell
Novell, Inc.
1555 N. Technology Way
Orem, UT 84057.2399
(801) 222-6000
Fax: (801) 861-3933
http://www.novell.com/

IRS

4 13 Very Useful
4 14 Moderately Useful
415 Not Useful

Jonathan Miller

Selecting Analog I/O Boards for Embedded Systems

Real-world problems don't come in binary mode. So our problems can get a computer solution, Jonathon shows us how to match the resolution, sample rates, input range, and so on of our problem with the right board.

Embedded computer systems find applications in all sorts of places that require the computer system to interface to the real world in some way. It might be in a space-shuttle botanical monitoring facility, a grocery-store checkout scanner, or a weather balloon. The interface can be either high or low level.

In a high-level interface, the data is already in computer-readable form. With a checkout scanner, for example, the computer receives data from an optical scanner module, which outputs a serial datastream. Minimal, if any, hardware engineering effort is required to interface the two products together.

A low-level interface, on the other hand, handles raw data or signals in analog or digital form. This data usually cannot interface to a computer's standard I/O ports. Instead, it requires an intervening data-acquisition board and additional software.

In a weather balloon, for instance, the signals read by the computer could include analog voltages from temperature and humidity sensors, as well as a pulse train from a rotary encoder indicating wind-speed.

Data acquisition is a wide field comprising several types of data, including analog, digital, and time-based digital (e.g., pulse trains or rotary encoder outputs). To keep this discussion manageable, I'll confine it to the realm of analog I/O.

The key component of all analog-input boards is the ADC, which converts the real-world analog signal to a digital number.

An ADC's resolution (i.e., the smallest input-signal change it can detect) is expressed as a fraction of its total input-signal range. Resolution is usually expressed as a binary value (e.g., an 8-bit ADC resolves an input signal to within 1/256 [$1/2^8$] of its full-scale input range).

Some ADCs (e.g., those designed for voltmeters) output BCD numbers. Resolution is expressed in digits (e.g., 4.5 or 6 digits).

A DAC operates on the same principles. The digital number you write to the chip determines its analog output voltage.

Analog I/O is an area of special significance for embedded systems. The most recent *PC/104 Resource Guide* lists 26 companies that make PC/104 analog I/O modules. Tables 1 and 2 detail the key features of PC/104 analog I/O modules from five popular companies.

Table 1 lists top-of-the-line 16-bit resolution modules, representing the leading edge in PC/104 data acquisition. Table 2 shows the more common 12-bit modules that fit a wider range of applications and budgets.

RESOLUTION

The most common resolution on the market today is the 12-bit ADC since it is adequate and cost effective for the bulk of real-world applications. Of course, applications like hi-fi audio require 16-bit ADCs, and some low-resolution applications like weather indicators may only require 8 bits of resolution (i.e., no one really cares if the temperature outside is 71° or 71.238°).

As you'd expect, higher-resolution ADCs cost significantly more than lower-resolution

Company	Analogic	ComputerBoards	Diamond Systems	Real Time Devices	WinSystems
Model	AIM-16-1/104	PC104-DAS16JR/16	Diamond-MM-16	DM5416-2	PCM-A/D16
Single-ended inputs	16	16	16	16	16
Differential inputs	8	8	8	8	6
Max. sample rate (kHz)	100	100	100	100	33
Max. unipolar range (V)	0–10	0–10	0–10		0–5
Min. unipolar range (V)	0–1.25	0–1.25	C-l .25	–	c-5
Max. bipolar range (V)	±10	±10	±10	[illegible]	±10
Min. bipolar range (V)	±1.25V	e0.625	e0.625	±01.25	±10
Programmable gain	Yes: 1, 2, 4. 6	Yes: 1, 2, 4, 6	Yes: 1, 2, 4, 6	Yes: 1, 2,4, 6	No
A/D trigger options	S/W, pacer clock, external	S/W, pacer clock, external	S/W, pacer clock, external	S/W, pacer clock, external	S/W
A/D transfer to memory	S/W, Interrupt, DMA	S/W, Interrupt, DMA	S/W, Interrupt DMA	S/W, DMA	S/W, Interrupt
Counter/timers	1 24-bit A/D pacer clock	1 32-bit A/D pacer clock. 1 16-bit general purpose	1 32-bit AID pacer clock, 1 16-bit general purpose	1 32-bit A/D pacer clock, 1 16-bit burst timer, 16-bit sample counter, 2 16-bit general purpose	0
Analog outputs	0	0	4	1	0
D/A resolution			12 bits	16 bits	–
Max unipolar range			0–10V		–
Min unipolar range			o-5v		–
Max bipolar range			±10V	±10V	–
Min bipolar range			±5V	±10V	
Simultaneous update	–		Yes	Yes	–
DIO	16 DIO	4 in. 4 out	6 in, 8 out	16 DIO	0
Programmable DIO direction	Yes	NO	NO	Yes	
DIO state on powerup	All lines in input mode	outputs set to 0	outputs set to 0	All lines in input mode	
Operating temperature (°C)	+5 to +50 Std. –40 to +85 Opt.	0 to 50 Std. -25 to +60 Opt.	0 to 70 Std	0 to 70 Std. -40 to +85 Opt.	0 to 70 Std.
Power-supply voltages	+5	+5	+5	+5	+5, ±12
Power-supply current (mA)	500	250	175	200	150. ±50
Comments, special features	256-sample A/D FIFO. I/O connector not in PC/104 position	256-sample AID FIFO	Only 16-bit PC/104 module with multiple analog outputs	1024-sample A/D FIFO, programmable channel-scan seq	Low cost, A/D only
Driver software included	Yes	No	Yes	Yes	Demo only
Price (qty. 1)	$625	$499	$525'	$7952	3753

Notes:
[1] Without analog outputs, cost is $469.
[2] DM5416-1 is the same board without analog outputs at $695.
[3] The system with DC-to-DC converter for +5-V only operation costs $450.

Table 1: These 16-bit modules represent the state of the art in PC/104 analog I/O.

ones. For example, the Diamond Systems 8-bit ADC costs around $5, the 12-bit ADC is priced at $17, and their 16-bit ADC is available for $50.

In addition to the higher chip cost, PCB design becomes more critical with increased resolution. As well, the analog components surrounding the ADC have to perform to the same level of precision.

It therefore makes sense that 16-bit boards cost substantially more than 12-bit ones. For cost-conscious applications, analyze the data resolution you need and choose your analog-input board appropriately.

SAMPLE RATE

Sample rate indicates how fast the ADC can update its input-signal measurement. The higher the input signal's rate of change, the faster the ADC must be to keep up.

Most real-world applications have slow-changing signals (e.g., l-10 Hz). Since the typical analog-input board samples at -100 kHz, this rate is quite slow.

Certain types of inputs (e.g., audio signals or strain gauges) require sample rates approaching the upper limits of these boards. Some applications, like digital oscilloscopes, sample at rates of 100 MHz to 1 GHz.

However, the typical 30-l 00-kHz analog-input board is adequate for most applications. The variations in maximum sampling rate shown in the tables are probably not important.

There's one more detail that affects sample rate-the number of input signals being monitored simultaneously. An ADC's required sampling rate goes up in proportion to the number of signals you have.

MULTIPLE INPUT RANGES

Most analog-input boards today offer multiple input ranges so you can tailor the ADC's input range to the signal range and maximize the resolution of your readings.

If an input signal's range is O-l V and the analog-input board's range is O-l 0 V, you only get a tenth of the board's useful range. So, resolution is only a tenth of what it could be.

This may not be an issue if your board has a higher resolution than you need. In general, however, you're better off with a board whose input range can be adjusted to match your requirements.

PROGRAMMABLE GAIN

Since there's a wide variety of input signal sources, it's often the case that an application has input signals with different ranges. In this situation, programmable gain or programmable input range lets the software change the board's input range in real time to match the range of each input channel individually.

On boards without programmablegain, all inputchannels have the same range all the time until you physically change the board with a jumper or switch. If you have a wide range of input signals, you should consider programmable gain.

Gain and input range are essentially reciprocal to each other. The input range specifies how wide a range of input voltage the ADC can read without reaching its limit, while the gain is simply the ADC's input range divided by the input signal range:

$$\text{Input signal range} = \frac{\text{ADC input range}}{\text{Gain}}$$

$$\text{Gain} = \frac{\text{ADC input range}}{\text{Input signal range}}$$

For example, Diamond System's Sapphire-MM analog I/O module (see Table 2) uses an ADC with an input range of ±5 V in bipolar mode. To support different input-signal ranges, it has a front-end programmable-gain amplifier with gain options of 1, 2, 4, and 8 (binary) or 1, 10, and 100 (decimal).

For an input signal with a range of ±0.5 V, select a board with decimal gains and amplify the input signal by 10. Even though the input signal swings from -0.5 to +0.5 V, the ADC still sees a signal ranging from -5 to +5 V.

ADC TRIGGERING AND TIMING

There are several ways to initiate an A/D conversion. A software command to the board is simplest since it usually requires the least amount of programming and hookup. Software-triggered conversions are adequate for applications where

Table 2: There **12-bit** PC/I 04 analog **I/O** modules fit a wide range of applications well.

Company	Analogic	ComputerBoards	Diamond Systems	Real Time Devices	WinSystems
Model	AIM-12-I/104	PC104-DAS16JR/12	Sapphire-MM-12AB	DM5406-2	PCM-AIO
Single-ended inputs	16	16	8	16	8
Differential inputs	8	8	8	8	8
Max. sample rate (kHz)	100	160	100	100	30[1]
Max. unipolar range (V)	0–1 O(3)	0–10		0–10	0–5
Min. unipolar range (V)	0.1	0–1.25		C-I .25	0–2.5
Max. bipolar range (V)	±10 (3)	±10	±5	±10	±2.5
Min. bipolar range (V)	±0.1	±0.625	±0.5 or ±0.625	±0.625	±2.5
Programmable gain	Yes: 1, 10, 100	Yes: 1, 2, 4, 8	Yes: 1, 2, 4, 8 or 1, 10, 100	Yes: 1, 2, 4, 8	NO
A/D trigger options	S/W, pacer clock, external	S/W, pacer clock, external	S/W, pacer clock,' external	S/W, pacer clock, external	
A/D transfer to memory	S/W, Interrupt, DMA	S/W, Interrupt, DMA	S/W,Interrupt, DMA	S/W, DMA	S/W, Interrupt
Counter/timers	1 24-bit A/D pacer clock	1 32-bit A/D pacer clock, 1 16-bit general purpose	1 32-bit A/D pacer clock, 1 16-bit general purpose	1 32-bit AID pacer clock, 1 16-bit general purpose	0
Analog outputs	0	0	2	2	2
D/A resolution (bits)	–		12	12	12
Max unipolar range (V)	–		0–10	0–10	0–5
Min unipolar range (V)	–		0–1	0–5	0–5
Max bipolar range (V)				±5	±5
Min bipolar range (V)			–	±5	±5
Simultaneous update	–	–	Yes	Yes	No
DIO	16 DIO	4 in, 4 out	4 in, 4 out	16 DIO	0
Programmable DIO direction	Yes	NO	NO	Yes	
DIO state on powerup	All lines in input mode	Outputs set to 0	Outputs set to 0	All lines in input mode	
Operating temperature (°C)	+5 to +50 Std. -40 to +85 Opt.	0 to 50 Std. -25 to +60 Opt.	0 to 70 Std.	0 to 70 -40 to +85 Opt.	0 to 70
Power supply voltages (V)	+5	+5	+5, ±12	+5, ±12	+5, ±12
Power supply current (mA)	500	250	175, ±36	175, ±30	30, ±25
Comments special features	256-sample A/D FIFO, I/O connector not in standard pos.	256-sample AID FIFO	Wide variety of configurations available to match application	1024-sample A/D FIFO, programmable channel-scan sequencer	LOW power
Driver software incl.	Yes	No	Yes	Yes	
Price (qty. 1)	$423	$399	$325[2]	$4393	$295'

Notes:
[1] Company's stated max rate is 100 kHz. However, no DMA function is provided for high-speed data transfer, which would be required to achieve this rate. The rate listed in the table is a typical maximum for interrupt operation, which is supported by the board.
[2] Ten models are available ranging from $195 to $375.
[3] RTD's DM5406-2 is the same product with +5-V only operation at $539.
'Without analog outputs, cost is 5250.

samplingisdoneatalowspeed or where the time between samples is not critical.

An onboard A/D pacer clock (i.e., a counter/timer combined with a clock oscillator) is useful for higher-speed conversions. A high-frequency oscillator isconnected to a counter/timer circuit with a programmable divisor value. Changing the divisor alters the output frequency of the pacer clock.

By using the pacer clock to control the ADC, A/D conversions can occur automatically atexactlythe programmed rate without further software intervention. Most of the boards in Tables 1 and 2 have counter/timers in varying configurations to enable automatic high-speed sampling.

Pacer clocks are not just for high-speed sampling. The large width of the pacer clock means you can have large divisors for correspondingly slow sample rates.

Both the ComputerBoards PC 104-DAS16JR/16 and the Diamond Systems Diamond-MM-16 provide a 32-bit pacer clock with a l-MHz oscillator. Sample rates can range from the board's maximum of 100 kHz to its minimum of 0.0002 Hz. Photo 1 shows the PC 104-DAS16JR/16, and Photo 2 is of the Diamond-MM-l 6.

Many applications need A/D conversions to synchronize with real-world events. To take samples every time a wheel reaches a certain point, you need an external hardware input signal to connect to the board. A good data-acquisition board offers all three types of A/D conversion control.

Triggering is more complex. Don't confuse the term "trigger" with "timing" or "clocking" as they are very **different. Tim**ing controls the sample rate (e.g., one conversion every millisecond), while triggering starts an entire conversion sequence (e.g., sample an input channel at 100 Hz for 0.5 s when a digital input line goes high). Many data-acquisition boards provide external digital triggering as well as external clocking.

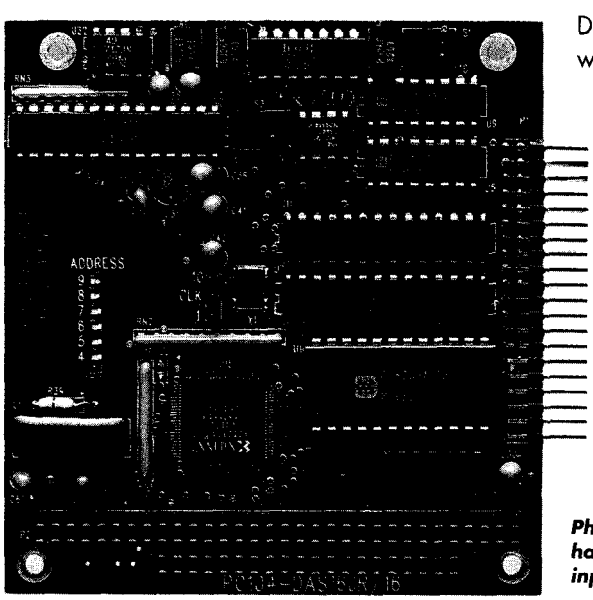


Photo 1: The PC104-DAS16JR/16 has 16 characters of 16-bit analog input but no analog outputs.

Frequently, there are hidden features or limitations on various methods of A/D conversion timing and triggering, as well as alternative ways to achieve the desired behavior. If your application involves tricky timing, discuss the details with an applicationsengineer prior to purchasing the board.

GETTING DATA TO MEMORY

Once the ADC converts the input signal to a numeric value, the CPU must have access to it. The simplest way is to read the data from the chip.

All thedataacquisition boards reviewed here use I/O addressing to read the values stored in the ADC. The application program simply reads 2 bytes from the board and assembles them into 12- or16-bit data.

For high-speed applicationsor multitask real-time situations, data transfers in the background. On the PC/I 04 bus, the two methods for background transfer are interrupts and DMA.

An interrupt routine transfers data from the board to the system memory independent of main program flow. Each A/D conversion generates an interrupt request.

The CPU executes the designated routine, **which reads the data from the board,**

performs housekeeping functions, and may execute other functions, such as storing the data in a memory buffer for access by the main program.

Interrupts are great for getting data at speeds of up to -35,000 samples per second (depending on CPU speed). But, they have two main drawbacks. The routine requires multiple CPU cycles to execute, taking a significant amount of time. And, there's usually a built-in delay in response to interrupt requests, which limits the maximum interrupt rate.

DMAs transfer data from the board to memory in a single cycle, so no software interference is necessary. In fact, DMA cycles occur in between CPU instructions, with no interruption in program flow and no knowledge by the program that they occurred.

WHAT ABOUT FIFOs?

An onboard FIFO buffer enhances interrupt or DMA operation. It enables the CPU to be decoupled from the board's operation, so it doesn't have to respond every time a new A/D conversion occurs.

At high sample rates in complex operating environments like Windows, this feature is necessary because it's impossible to guarantee that Windows will respond quickly enough to get the data from the board before a new sample wipes it out. With a FIFO, the board can sample at the desired rate, and Windows can respond less frequently, transferring a block of data at once.

Conversely, at low sample rates or in less complex software environments, a FIFO only adds to the board's cost. It's standard practice to get 100,000 samples per second under DOS without a FIFO. So, **unless** you're using Windows, you probably don't need one.

CHANNEL SCANNING

Real Time Devices provides this unique hardware feature in their DM5416-216-bit datamodule shown in Photo 3. It enables you to program a sequence of channels to convert in random order and at varying sample rates. Because the channel-gain function is implemented in hardware, you can **randomly** switch channel gain at the full conversion rate of the board.

Channel-scan sequence programming is useful in specialapplications. Since every board lets you change the input channel setting in software, you can create the same effect. However, code limits you to a slower rate and introduces jitter into the sample time.

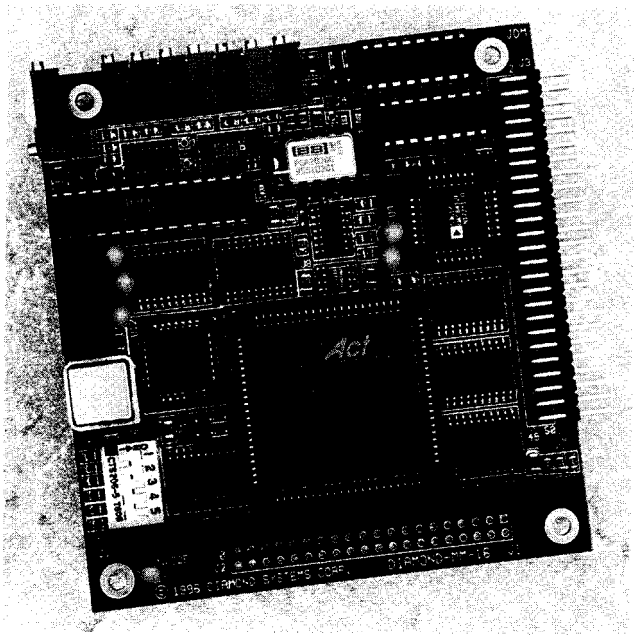


photo 2: The Diamond-MM-16 ***PC/104*** analog ***I/O*** module provides 16 channels of ***16-bit*** analog input and 4 channels *of* ***12-bit*** analog output. It *requires only* ***+5-V*** *power* supply and operates over o -25 to ***+85°C*** temperature range.

ANALOG OUTPUT SPECS

In the real world, input is needed the most, but many applications also need analog outputs. Why, then, do most manufacturers leave analog outputs off their high-end boards?

The answer isn't board space. Both Analogic and ComputerBoards have enough room on their boards to accommodate the components. Who knows the real reason, but makers of boards with both input and output (e.g., Diamond Systems and Real Time Devices) are happy to take advantage of this oversight.

One big difference between ADCs and DACs is that, although it's simple to add an input multiplexer in front of an ADC to provide multiple input channels, the same is not true for DACs. It's possible to multiplex a single DAC onto several output channels, but complicated, expensive circuitry is required.

Therefore, virtually all analog output boards use an independent DAC for each output channel. However, many ICs contain more than one DAC per chip, so one chip may serve several channels.

Using independent DACs drives up the cost of the board and explains why analog output costs more per channel than analog input. But, the relative price of the high-end boards still doesn't explain the absence of onboard analog outputs!

OUTPUT RANGES

Unlike ADCs, which have fixed input ranges, by adjusting reference voltage(s), DACs can usually have multiple output ranges. A DAC's output voltage is equal to its reference voltage times the ratio of the input data to the maximum possible input value. Therefore, the output voltage can be adjusted both with a digital numeric value and an analog voltage.

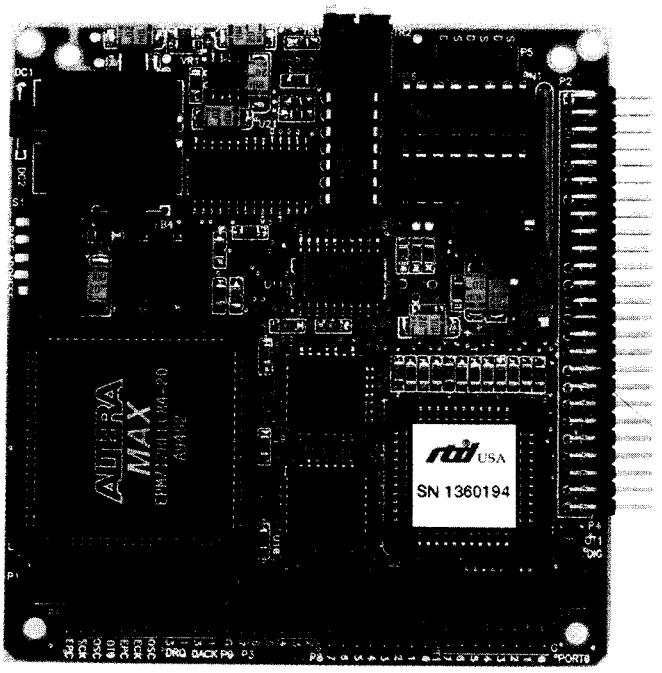


Photo 3: *Real Time Devices'* DM54 16-2 is a ***16-bit*** data-acquisition board with random ***channel*** gain *scan* memories and ***FIFO*** buffers.

The DAC can then be used as a digital attenuator by feeding the signal into the reference voltage pin and writing values to the DAC to control the output level.

SIMULTANEOUS UPDATE

In many applications, you need to update multiple channels at exactly the same time. For example, lasers in a light show are driven by DACs—one controlling the x position and the other the y position.

If the x and y positions are not set at exactly the same time, the laser traces a jagged stair-stepped shape instead of a straight line connecting the old and new positions. A good analog outputcircuitcan simultaneous update across all channels.

OTHERSPECS

Power consumption is an issue in many embedded systems. Pay attention to the power-supply voltage and current requirementsofthe **boards you** select. But, this isn't as simple as it seems.

A board that operates on +5 V sounds ideal at first since you only need a single output on your main power supply. However, the analog I/O boards only requiring +5 V use miniature DC-to-DC converters to supply power (usually ±15 **V) to the** analog circuitry.

These converters supply wider operating voltages so the analog circuitry can handle signals of up to ±10 **V-a** range that isn't possible on a single +5-V supply.

However, the cost of these converter circuits may approach or even exceed the cost of supplying the additional voltages, especially if you use more than one board in your system.

Operating temperature is another important issue. Since many embedded systems are used on vehicles, they're exposed to wide temperature ranges. The lower end of the board's range is usually more significant, especially for airborne vehicles.

Most manufacturers do not guarantee all specs of their boards over the entire temperature range-only that the board operates over this range. Specifications are usually tested and/or guaranteed only at 25°C.

Software drivers (or function libraries) can becritical toanyapplication'ssuccess. For simple, low-level operations, drivers may not be necessary since direct programming of the board is often as easy as learning the manufacturer's software.

However, for complex operations, the driver saves a lot of time. All the manufacturers listed in this article with one exception provide free drivers and sample programs with their boards. ComputerBoards provides a free installation and calibration program but sells a separate programming library for $49. PCQ.EPC

Jonathan Miller is president of Diamond Systems *in Palo Alto, CA, a founding* member of the *PC/104* Consortium and the first company to make a line of data-acquisition *modules for PC/I 04. Previous/y, he worked in technical sales and design engineering at MetraByte and Cyborg, two early manu*facturers of PC-based data-acquisition boards. You may reach him at *jdm@ diamondsys.com.*

SOURCES

AIM-16-1/104, AIM-12-1/104
Analogic, Inc.
360 Audubon Rd.
Wakefield, MA 01880
(508) 977.3000
Fax: (617) 245-I 274
dcpinfo@analogic.com

PC104-DAS16JR/16, PC104-DAS16JR/12
ComputerBoards, **Inc.**
125 High St., Ste. 6
Mansfield, MA 02048
(508) 261-1 123
Fax: (508) 26 I-1 094
info@comp-boards.com

Diamond-MM- **16,** Sapphire-MM- **12AB**
Diamond Systems Corp.
A50 San Antonio Rd.
Polo Alto, CA 94306
(415) 813-I 100
Fax: (415) 813-I 130
techinfo@diamondsys.com
http://www.diamondsys.com/

DM5416-2, DM5406-2
Real **Time Devices, Inc.**
200 Innovation Blvd.
State College, PA 16803
(81*A)* 234.8087
Fax: (814) 234.5218
sales@rtdusa.com

PCM-A/D16, PCM-AIO
WinSystems, **Inc.**
715 Stadium Dr.
Arlington, TX 7601 1
(8 17) 274.7553
Fax: (8 17) 548-I 358
info@winsystems.com

PC/I 04 Consortium
P.O. Box 4303
Mountain View, CA 94040
(415) 903-8304
Fox: (415) 967-0995
pcI 04@ix.netcom.com

IRS

416 Very Useful
417 Moderately Useful
418 Not Useful

APC

Applied PCs

Fred Eady

Vinegar or CLampagne?

BBS Software Managing Remote Data Acquisition and Control

Fred takes us to the land of remote data acquisition and control using Wildcat! BBS software and Datalight's ROM-DOS. After proving you don't need a desktop PC as a server, he rigs the BBS to do the data acquisition and control.

Dang it...it's dark out here! I squinted my eyes and reached for my partner in crime. "Mark, hand me the flashlight."

A beam of high-intensity light cuts through the warm Florida night. "Got it. Did you bring the telephone?" I whispered. "Now. I thought you had it."

Now what? We stand here in the dark, wondering how to run this test without a phone. Attempting not to lose face, I dig through my tool bag. There must be something I can use.

Aha! My trusty line tester! An RJ-11 equipped cable, too! A flip of the power switch and behold--LED city.

I turned to Mark, tester in hand, "Ha! We've got it beat now! Hey, hold this light." He illuminated the test jack as I unplugged the feed and inserted my tester cable. I applied a 600-Q termination-no dial tone.

"Shoot, looks like the incoming line is dead." "OK, now what?" asked Mark.

"No problem. Necessity is the mother of @y-rigging. I have some really long extensions. We'll run one from the kitchen jack and use the fax and Internet lines for the other two. That'll get us the lines we need."

Mark beamed, "Make it so, Number 1." A couple hours and a few cable runs later, the framework of this article was in operation.

Mark and I frequently role play as Jean-Luc and Riker (no, we're not over the deep-space end yet). But this time, we were creating the new lifeform. We were going where no embedded application had gone before-without joining the fleet or getting our molecules scattered about the heavens.

ERDAC

ERDAC is short for Embedded Remote Data Acquisition and Control. I'll stick my neck out and prognosticate (that's Southern for "assume") that 99.9% of you have logged on to some sort of BBS at least once.

Most of you probably use a BBS or other dial-up services on a regular basis. You dial, log on, download, upload. No biggie.

You've probably become numb with accessing data, chatting with colleagues, and so on. Matter of fact, you're probably slapping this magazine on your leg saying, "I don't believe *INK* is letting Fred do an *EPC* article on how to use a BBS!"

To quote the material girl [and new daughter), "Not!" ERDAC is a unique BBS-based combination of embedded and some not-so-embedded hardware and software components that gives one or more remotely monitored or controlled embedded systems access to a variety of data-acquisition hardware and software.

If you need to implement an embedded messaging system or traditional BBS, read between the lines and you'll **see you** can do that, too. But, the goal today is to stuff a Wildcat! down the throat of a VIPer and remotely control some external hardware.

CHARACTERISTICS OF THE MUST

The ERDAC model requires a fast, stand-alone, full-function, AT-compatible, '486 in a small form factor. ERDAC hardware must be able to function without a keyboard, display, or mouse in harsh environments. If the power fails, the ERDAC must automatically reboot and reload the compressed BBS software and applications.

To access the sensory peripherals and provide the selected BBS system with an appropriate fail-safe operating environment, a minimum of two serial ports, a parallel port, flash memory, and an ample amount of RAM are required.

For automated operations, you need a battery-supported real-time clock. You also want provisions for an optional diskette and hard disk for development purposes.

OK. Those are the ERDAC requirements, and here are the reasons why. Two serial ports allow RS-232-based data-acquisition peripherals to report to the ERDAC on one serial port, while the BBS software communicates with the requester on the other port. Some peripherals use the parallel port for I/O, and it's also an excellent port for user-written control applications.

The flash memory is essential because it holds all the preconfigured ERDAC system files and any other files needing to be replenished on powerup. My ERDAC model won't include mechanical disk devices, although the VIPer could support them in this application if necessary.

The embedded system of choice for ERDAC is Teknor's VIPer806, shown in Photo 1. My '806 is equipped with all of the above plus.

Its flash memory size totals 4 MB with 16 MB of RAM onboard. My version of ERDAC stores the compressed BBS image in flash and installs it into a VDISK for execution on powerup.

JUST A DATA PICKER

Why use a BBS for data acquisition?

I liken it to using desktop PCs as embedded servants. All the stuff needed to complete the task is most likely already there.

That axiom certainly works for ERDAC's BBS-based software. Any run-of-the-mill BBS can inherently transfer files, provide a modem interface, and shell to DOS or a user-specified program. Why write and debug communications code when you can open a can of communications whip-butt and pour it into an embedded target?

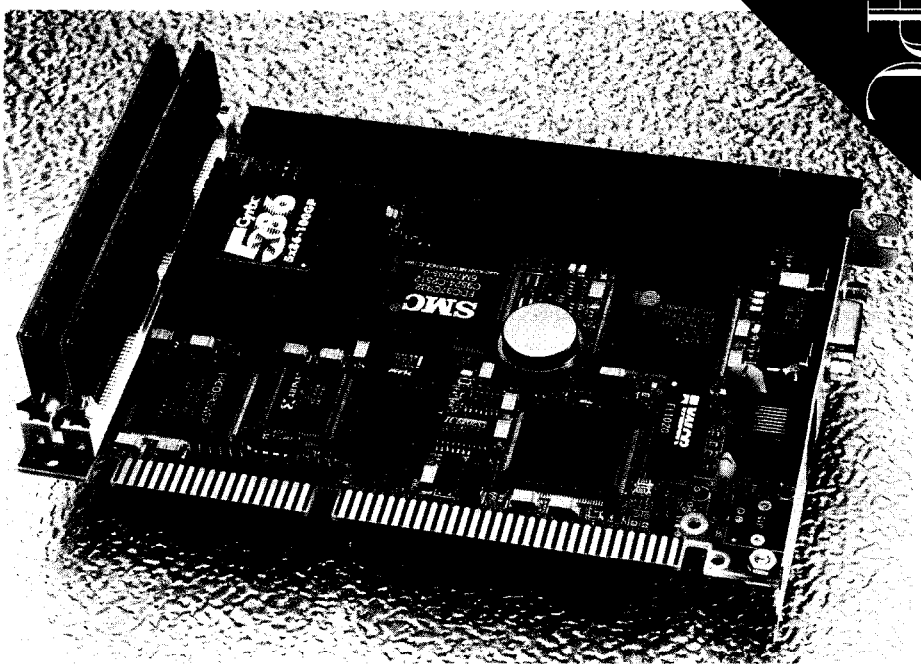


Photo 1: Here's o look at the snake that ate the wildcat.

A bunch of BBS programs out there do all sorts of amazing things. ERDAC only requires that the BBS software have the previously mentioned basics and be capable of running in a DOS environment.

That doesn't narrow it down any, so I'll make a decision. My software of choice for ERDAC is Mustang Software's Wildcat!. And, why did I choose Wildcat!? I'll be honest and up front.

I've used itforyearsas the EDTP BBS. It's inexpensive. It'seasytoadminister. Itworks. And, I have a "real" license for it. But....

Weighing in at over 15 MB in base mode and backed by a 0.878" instruction manual, the Wildcat! BBS software wasn't intended as an embedded application. Its enormous size implies enormous functionality. Wildcat! is everything a BBS can be.

The sysop has hundreds of software knobs to turn and can specify BBS access on the color of your underwear on a certain day at a certain time. Then there are file areas, conferences, questionnaires, user databases, fax processing, and such.

This BBS even processes Mastercard and Visa! Obviously, ERDAC doesn't need that level of security or complexity. And, if your embedded creation ever on its own prompts you for a plastique number, leave town. The aliens have discovered us.

As you read on, it'll become clear that the embedded ERDAC is basically an embeddable Wildcat! configuration created by stripping all unneeded functions from the base BBS package.

GETTING OAKEN FLAVOR

This is not your father's BBS. As I think about it, ERDAC is really a DTS (Data Transfer System), not a BBS.

Its mission is to provide access to data-acquisition hardware attached to the VIPer's serial and parallel ports with little or no interface code at all. (The data requester then has more time for writing in Steve's favorite language, "solder.")

Since all my selected data-acquisition and control hardware is smart and interfaces to ERDAC via DOS-based software drivers, ERDAC needs only to be able to establish a connection to the sensory peripherals and give control of the ERDAC system to the remote requester on demand.

It doesn't require the sensory devices to be smart. The dial-in user has control of ERDAC's software and hardware and can effect any operation on any sensor devices the DOS environment allows.

For instance, using DOS Debug, the remote requester can toggle or read the ERDAC parallel port bits or control the second serial port. In fact, the ERDAC system is configured by including or excluding functionality within the BBS and DOS environments according to your application requirements.

The trimmed-down Wildcat! BBS image inherently provides a means for the bidirectional transfer of files and data between the host requester and ERDAC. It also has a timed event trap that lets VIPer's hardware access the external sensory peripherals on

a scheduled basis without user intervention.

This batch-file-driven trap can retrieve and store data from devices such as intelligent data loggers for later transfer to a dial-up requester.

Most commercial communications packages have an executable scripting language that enables the implementation of automated operations between the requester and the selected ERDAC.

Note the word "selected." Since the system is dial-up, there can be as many remote ERDACs as the user requires. If your ERDAC node requirementsexceed the number of phone lines the phone company can provide, buy a ticket for the next Mars mission. You belong there.

Another built-in ERDAC plus is network access. If your ERDAC is attached to a network, it's a simple matter of dialing into the network-active ERDAC to access all the shared network resources. I have an ERDAC node configured to do just that.

When I'm on the road with the laptop, I always need a file on the computer without the modem in the shop. Conversely, ERDAC can also be configured to call home.

The final installed ERDAC image consists of the base Wildcat! code, software and hardware definitions, and control programs for the external hardware. Wildcat! base code is a combination of the Wildcat! executable coupled with all of the necessary housekeeping data files generated when the BBS is initially configured.

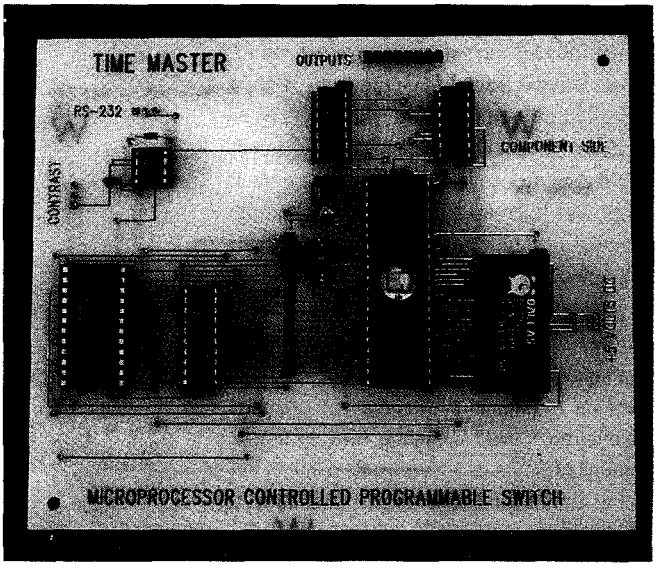


photo 2: *I'll* bet that old 8749 never thought it would be playing with stuff like ***EEPROMs, RAM***-laden real-time clocks, or power-stealing ***RS-232 ICs.***

General BBS information, file areas, and system security compose the software definitions. The hardware definitionscontain such things as modem and memory configuration information. My ERDAC system will have dominion over an intelligent valve system, so the controller application program (TIME - MAS1 . EXE) is also embedded with the ERDAC file image.

After removing the fat from the off-the-shelf Wildcat! BBS, the image is compressed via PKZIP for storage on the VIPer's ROM-Disk. My final ROMDisk-resident ERDAC BBS image size is less than 1 MB, including drivers and a DOS command processor!

A LITTLE WINE AND CHEESE

Since ERDAC is a low-fat embedded application, it makes sense to embellish it with other healthy ingredients. Datalight's ROM-DOS is a perfect complement to the reduced-weight Wildcat! file image.

Although ROM-DOS can be executed from ROM as a binary image or from a Datalight ROMDisk, the VIPer comes equipped with its own built-in ROMDisk technology. Therefore, I can use the familiar DOS-compatible file system instead of binary images. In other words, the VIPer and ERDAC use ROM-DOS right out of the box.

Oh, yeah. The answer is yes. Regular old MS-DOS would work for ERDAC. The advantage of Datalight ROM-DOS over the normal diskette DOS is cumulative file size. For example, Datalight's command processor is a little less than 27 KB deep versus almost 55 KB for Microsoft's version.

Even the Datalight drivers are lean—29KBforthediskette versionof HIMEM.SYS versus less than 3 KB for Datalight's. And, the two command processors are functionally identical!

ROM-DOS is a fully functional DOS complete with almost all the internal and external bells and whistles found in the diskette version. The only external DOS command I needed for ERDAC was XCOPY.

XCOPY lets me copy the ERDAC image into a VDISK directory structure without first having to build one. All other functions reside within the ROM-DOS kernel files IBMBIO.COM and IBMDOS.COM. Butter. Parkay. Butter. Parkay.

My VIPer806 contains 4 MB of flash memory-more than enough to hold the compressed ERDAC image. But, why waste space because you can? We're not writing desktop applications here, bub.

The conserved space leaves an opportunity to embed additional ERDAC functions you might need or think of later. This configuration only consumes a little over 2 MB of combined ROM and RAM space when fully deployed.

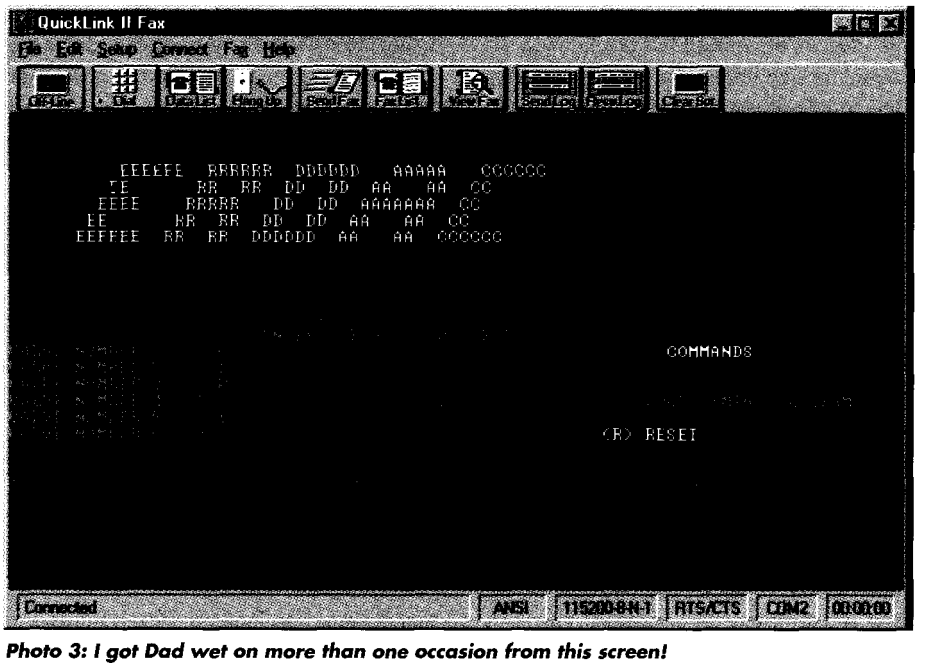
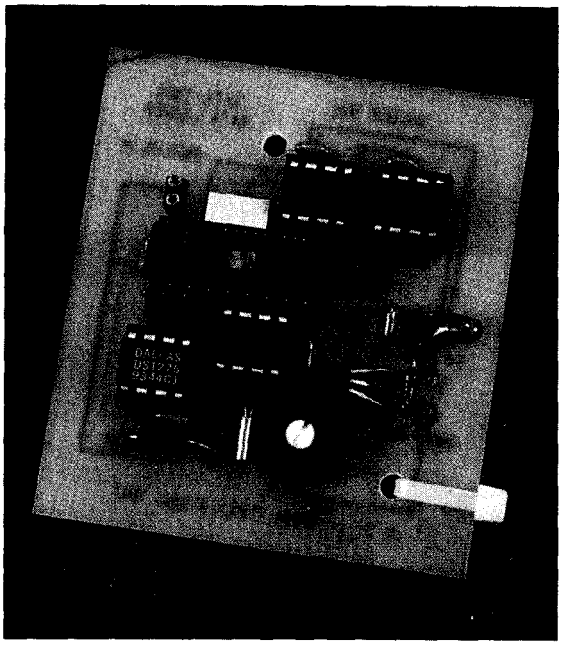


Photo 3: I got Dad wet on more than one occasion from this screen!

Photo 4: This simple *little datalogger interfaces to anything that can* generate voltages between 0 and *+5 VDC. It* can also *be programmed* A send an "I'm alive" *character* every 60 *s.*



DOORWAY TO THE CELLAR

DOORWAY is a BBSadd-in product from Byterunner. For the ERDAC application, it's the most important file in the group as it provides the logical path to the VIPer's peripheral hardware.

In BBS terminology, doors are exits to programs. When you enter a door, you find yourself in an independent application but still within the BBS shell.

I established our doorway to the external peripherals by first configuring a door menu item within the Wildcat! BBS image. This menu item points to a batch file I created called DROPDOS. BAT, which is located in the WI LDCAT\BATCH directory on VDISK.

Herearethecontentsof DROPDOS. BAT:

```
DOORWAY COM2 /S:*/G:ON/V:B^U
  /M:1000 /C:DOS
```

DOORWAY redirects all keyboard input and screen output through VIPer's COM2 serial port. COM2 connects the ERDAC and the data requester, freeing applications to access VIPer's COM1 **for their serial I/O purposes.**

The /S:* parameter says not to look for any specific files. Graphics and cursor control is turned on with /G:ON, and /V:B^U **indicates that BIOS redirection is enabled with the capability of switching to direct screen mode via a Ctrl-U key sequence.**

Therefore, programs using direct screen writes instead of BIOS calls can be viewed through the ERDAC requester's remote window. The final two parameters enable you to remain in the DOS shell for 1000 **min.**

Now, it's relatively simple to establish a communications link to ERDAC via our strip peddown BBS. Onceconnected to ERDAC, I enter a doorway to access the external peripherals attached to my VIPer806.

Now that you know the hows and whys of ERDAC, let's dial it up and control some water valves!

THE GRAPE'S BEHIND IT ALL

My family is in the business of growing grapes. Being the family's "computer type," **I was asked by my father to design an automated irrigation system for the vineyard. No problem.**

The result was an 8749-based **little ditty with a real-time clock and EEPROM data storage capable of controlling up to eight solid-state relays. The relays control water valves throughout the vineyard.**

I designed the valve controller with an RS-232 serial interface. The idea is to download a timing schedule into the controller's EEPROM, set the Dallas real-time clock IC, and let it rip.

Just in case things get out of control or a temporary change in the water schedule is needed, the user can instruct the controller to manually control the valves via a terminal. Photo 2 is a shot of the controller hardware, and Photo 3 is the manual-control screen as viewed remotely via ERDAC.

The valve controller also transmits valve status via RS-232 every 60 s. This information can be viewed remotely using my ERDAC configuration.

In effect, from a remote PC running a generic communications program that has established a logical dial-up session with ERDAC and the intelligentvalve controller, I can:

- set the controller time of day
- edit and load a controller timing schedule into EEPROM

- manually control any combination of eight relays
- check valve status

Now that everything (including the tomato plants) gets water on a regular basis, Dad's only problem is listening to Mom complain about the water bill. He's working on that, too. He's already asked about an ERDAC node for his well pump! Looks like rain to me.

A PRACTICAL PRESS

The phone company finally repaired the lines in the shop, and that's when the real crazy came out in me. Mark and I couldn't resist trying all kinds of hookups with the ERDAC.

Earlier, I hinted that ERDAC was a data-collection point, so we tied a smart data-logger (see Photo 4) to one of its serial lines. The dataloggerconsists of a microcontroller with a built-in ADC, serial EEPROM, a real-time clock, and a serial interface.

We kicked off the datalogger data-retrieval driver program on an hourly basis using the Wildcat! event-trap utility. The datalogger allows set-up and data-retrieval functions via its serial port.

It was a simple matter of constructing a batch file to automate the data-retrieval process. Once the datalogger goes into data-retrieval mode, it doesn't come out until a processor reset occurs. No sweat.

We used a VIPer parallel port I/O bit to reset the datalogger microprocessor. In less than an hour, we were measuring voltages at the remote ERDAC console in the shop!

As night fell, things got really wacky. I had this idea: Why not dial up an ERDAC and then have it dial another ERDAC and access the files on the second system? My bright idea had no practical use I could think of, but it was tricky.

So, I attached a modem to the first ERDAC's extra serial port and installed another ERDAC node in one of the desktops in the shop. It worked! Sorta.

The linkswere established, but mycomm software didn't let me see in real time what wascoming across from thesecond ERDAC node. I had to capture the screens and view them offline. It was worth a try!

CHAMPAGNE, ANYONE?

I know it seems strange to take a piece of code designed for a totally different purpose and twist it into a totally unrelated application, but that's what makes embedded system design so interesting.

Here, I took an unassuming BBS program, mixed in ROM-DOS, added some stuff, deleted some other stuff, and BAP! I created a useful embedded application, and I only wrote a very few lines of batch file code!

ERDAC's uses are limited only by your imagination. Chancesare, ifyoucan implement an interface between your specialized hardware and your embedded PC, you can use the ERDAC application to control and monitor your creation.

These examples of various implementations of the ERDAC system once again prove that it doesn't have to be complicated to be embedded. APC.EPC

A special duh-huh thank you to Jamie Ferrier at Datalight Technical Support for helping me see the light.

Fred Eady has over I9 years' experience as a systems engineer. He has worked with computers and communication systems large and small, simple and complex. His forte is embedded-systems design *and communications.* **Fred may be reached at** *edtp@ddi.digital.net.*

REFERENCES

Datalight, Inc., ROM-DOS V.6.22 Developer's Guide, Arlington, WA, 1995.

Mustang Software, Inc., Wildcat! 4 Sysop Guide, Bakersfield, CA, 1994.

Teknor Industrial Computers, Inc., *VIPer806'486/* 5x86 Sing/e *Board Computer Technical Reference Manual,* V.2.2, Boca Raton, FL, 1996.

SOURCES

ROM-DOS V.6.22, **ROMDisk**
Datalight, Inc.
307 N. Olympic Ave., Ste. 200
Arlington, WA 98223
(360) 4358086
Fox: (360) 435-0253
http://www.datalight.com/

Wildcat!
Mustang Software, Inc.
P.O. Box 2264
Bakersfield, CA 93303
(805) 873.2500
Fax: (805) 873-2599
http://www.mustang.com/

VIPer806
Teknor Microsystems, Inc.
7900 Glades Rd.
Boca Raton, FL 33434
(561) 8836191
Fax: (561) 8836690

IRS

4 19 Very Useful
420 Moderately Useful
42 1 Not Useful

DEPARTMENTS

MICRO SERIES

Joe DiBartolomeo

Standard's for Electromagnetic Compliance Testing

Radiated and Conducted Emissions Testing

Part 2 of 4

It's down to the nitty gritty of FCC and European tests. After a close look at the EMI receiver, Joe checks into the details of radiated and conducted emissions tests.

Radiated and conducted emissions emanating from electronic equipment is so common that everyone has experienced it. Usually, EM1 is no more than a nuisance, like the crackling on broadcast receivers or crosstalk on portable phones.

However, EM1 problems can be much more serious-even life threatening. Emissions from portable electronic devices (PEDs) like cell phones and laptops can cause serious interference to commercial aircraft electronic systems. Occurrences of EM1 caused by passenger carry-on PEDs is on the rise and of great concern to the commercial airline industry.

EM1 also causes much concern in the medical field. There are documented cases of medical devices ranging from anesthetic-gas monitors to motorized wheelchairs suffering performance degradation due to radiated or conducted electromagnetic emissions. The problem is so widespread that many hospitals ban the use of cell phones.

Perhaps the most tragic case of EM1 occurred during the Falklands war when the H.M.S. Sheffield was sunk by an Exocet missile. The onboard radar that could have detected the missile was turned off because it inter-

	Frequency band		
	A	B	C & D
Characteristic	9-150 kHz	0.15–30 MHz	30-1000 MHz
Bandwidth at the -6-db points (kHz)	45	9	120
Charge time constant (ms)	500	1	1
Discharge time constant (ms)		160	550

Table 1—*These fundamental characteristics of a quasipeak receiver are taken from the 1993 CISPR 16-I. Note that the charge time constant is much faster than the discharge time constant.*

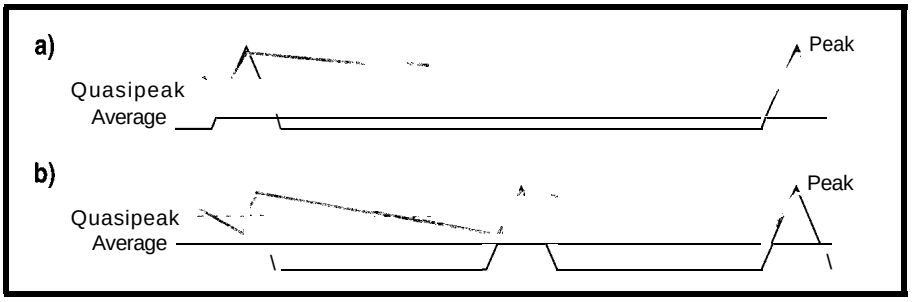


Figure *I--This graph shows the response of the average, peak, and quasipeak defectors to a repetitive signal with varying frequency. The solid blue line is the instantaneous response of quasipeak, and the dashed blue line is fhe quasipeak reading.*

fered with the ship's satellite communications system.

As the EM spectrum becomes more congested with more users in a fixed amount of spectrum, interference is bound to increase. I think it was Will Rogers who said, "Buy land. They don't make it anymore."

The same can be said for the EM spectrum. Designers of electronic equipment must be concerned about the EM environment their products will be subjected to from both an emissions and susceptibility perspective.

Regulatory bodies such as the FCC set up standards and tests that establish the EM environment and act as design standards for electronics designers. As discussed in Part 1, I'm concentrating on standards and tests that apply to equipment containing some form of digital circuitry.

The FCC and their European counterparts establish tests for radiated and conducted emissions levels for Class A and Class B digital devices.

Unfortunately, digital devices are quite good at producing both types of emissions. To see why, let's assume that the circuit board traces and wires present in digital devices can be modeled as simple dipoles.

The maximum radiated field occurs when the length of the antenna is ½λ (see equation 1). In other words, each frequency has a single antenna length (wire or trace) of ½λ that allows maximum radiated RF:

$$\lambda = \frac{c}{f} \qquad (1)$$

where λ is the wavelength in meters, c is the speed of light (i.e., 300 x 10^6 m/s), and *f* is frequency in cps.

An antenna of a length other than ½λ still radiates RF energy but with reduced efficiency. This reduction is normally quite large.

Let's apply to an antenna an EMF waveform that is a pure sine wave of frequency *f*. The antenna radiates energy at frequency *f*. To get maximum radiated RF energy, choose the antenna length to match the frequency. From equation 1, you see the antenna length should be $2c/f$.

Now, let's examine what happens when a square wave of frequency f is passed through a length of antenna tuned for *f*. The frequency f is radiated with maximum efficiency.

The frequencies at $3f, 5f, 7f$, and so on would also radiate (not, of course, with the same efficiency as *f*, but they would radiate).

For a pure sine wave or square wave, emissions could be varied by changing the antenna length. In fact, by knowing the frequency of the sine wave or square wave, you can select a proper wire or trace length to reduce emissions.

The problem with digital signals is that they are not pure square waves but trapezoids. David Prutchi explains this well in "Sniffing EM1 in the Near Field" (INK **71).**

For a trapezoid, it's the rise time that's important. The faster the rise time of the switching currents, the fuller the frequency spectrum. Recall that a dirac function has zero rise time and infinite frequency content.

When a digital signal with its full frequency spectrum is applied to a wire or trace, the probability is quite good that the digital signal contains a frequency that allows for a ½λ match.

Therefore, at some frequency (determined by the wire or trace length), there will be maximum radiated RF. Unlike the square wave or sine wave, altering the wire length only changes the frequency at which ½λ applies, which is what makes digital signals so prone to emitting RF.

Digital devices are also quite good at conducting RF energy along power lines. In a mixed analog and digital design, keep grounds separate to avoid the digital switching spikes from being coupled into the analog circuits.

A good design ensures that all digital noise is routed to the power supply. The power supply then conducts that RF noise out through the equipment and onto the AC lines.

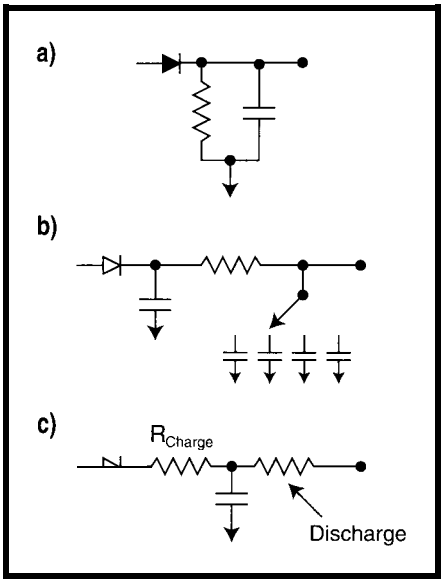


Figure 2—*Compare the simplified schematics of peak, average, and quasipeak defectors. As you can see, the time constant of the average defector can vary, and in fhe quasipeak defector, the discharge resistor is much larger than the charging resisfor.*

This, of course, is the problem. RF noise is transferred on the power lines to other devices on the power lines.

TESTING

When it comes to radiated and conducted emissions testing, the U.S. and Europe agree on the appropriate tests. The U.S. FCC Part 15 Subpart B and the European EN 50022 are the standards most often applied to digital devices. FCC Part 18 and the European Standard EN 50011 are also applied but less frequently, since they concern industrial, scientific, and medical devices.

FCC standards are based on ANSI 63.4, while European standards are based on CISPR 11, 16, and 22. But, the tests required by these standards are similar enough to be grouped. (CISPR 16 defines the specification for EMI test equipment.)

Radiated and conducted tests call for test equipment not commonly used by electronics designers. So, before discussing the tests, I'll review the equipment used to perform them.

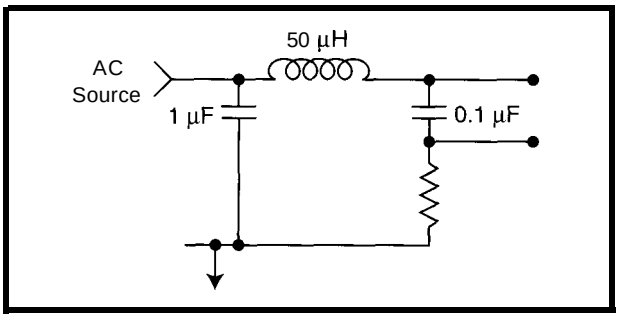


Figure 3-A Line Impedance Stabilization Network (LISN) isolates the equipment under test and filters out noise.

EMI RECEIVER

The most important piece of test equipment for both radiated and conducted emissions is the EM1 receiver. The receiver is essentially a spectrum analyzer with added functionality for EMI-compliance measurements.

Although a spectrum analyzer offers a quick check, it can't replace an EM1 receiver. EM1 receivers have superior sensitivity, dynamic range, and less susceptibility to overload.

More importantly, an EM1 receiver has a detector-the quasipeak detector-not found on the ordinary spectrum analyzer. This detector is mandated for tests involving digital signals.

The EM1 receiver has two other detectors-peak and average. They are straightforward, but the quasipeak detector requires some explanation.

The quasipeak detector was developed due to the limitations of peak and average detectors in measuring signals with repetition rates (i.e., digital signals). For example, if the peak detector measures a repetitive signal, the measurement is too high since no account is made of the fact that the signal switches on and off.

Likewise, if the average detector measures a repetitive signal, it doesn't reflect the fact that a peak value larger than the indicated average value is being emitted.

The quasipeak detector also mirrors the human response to pulse-type interference. With radio reception, for example, annoyance increases as pulse-repetition frequency (PRF) increases.

The quasipeak detector is weighted so that, as PRF increases, the value of the quasipeak detector approaches the peak-detector value. At lower PRF, the perceived annoyance decreases as does the quasipeak detector value.

Weighting the quasipeak detector is accomplished via charge and discharge time constants. Table 1 gives the specifications for the quasipeak detector. Figure 1 shows the response of the two detectors to signals with varying PRF, and Figure 2 shows their schematics.

Understanding how the quasipeak detector works is very important if your product is close to passing.

For example, let's say you're failing the radiated or conducted emission tests by 3 dB. You can then try to exploit the characteristics of the quasipeak detector. Depending on your product, you could use a clock with a less than 50% duty cycle, reduce your clock rate, or use slower logic.

Prior to attempting any of these tricks, study the quasipeak detector. Also, keep in mind that the standards

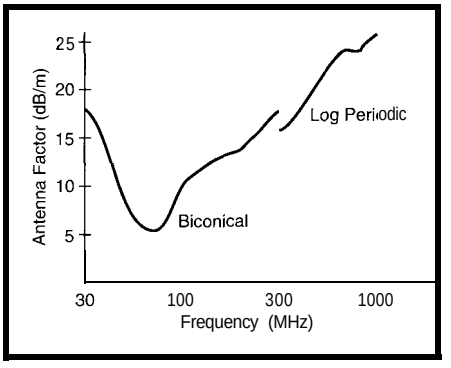


Figure 4-The Antenna Factor *(AF) is used* for biconical and log-periodic *antennas. It converts the received signal in dBμV/m into dBμV and then 50 Ω. The AF varies greater with increased frequency.*

people considered these things when they developed the quasipeak-detector specifications.

LISN

The Line Impedance Stabilizing Network (LISN), or artificial mains as it's sometimes called, performs several tasks. First, it isolates the equipment under test (EUT) from the AC mains.

The LISN filters out any noise on the incoming AC and couples the noise produced by the EUT to the EM1 receiver. It also filters any noise on the outgoing AC lines. The LISN presents a constant 50 Ω to the EM1 receiver.

CISPR 16 defines the LISN's specifications. Figure 3 shows a schematic of the LISN.

ANTENNAS

The receiving antennas are normally the biconical antenna for the frequency range 30-300 MHz and a log periodic for frequency range of 300–1000 MHz. When there is occasion to go above 1 GHz, the log periodic is usually used.

The test limits for radiated emissions are specified for the electric field strength in volts per meter, while the EM1 receivers are calibrated in volts into a 50-R load. The antenna must therefore be calibrated in terms of V_{out} into 50 Ω for a given field strength. This task is done with the antenna factor.

Each calibrated antenna comes with a chart showing antenna factor versus frequency. Figure 4 shows typical antenna-factor charts for both biconical and log periodic.

Equation 2 shows the relationship between volts at the EM1 receiver (V), the antenna factor *(AF),* the electric field at the antenna (E), and the cable attenuation *(A).*

$$V(dB\mu V) = E\left(\frac{dB\mu V}{m}\right) - AF\left(\frac{dB}{m}\right) - A\left(\frac{dB}{m}\right) \quad (2)$$

RADIATED EMISSIONS

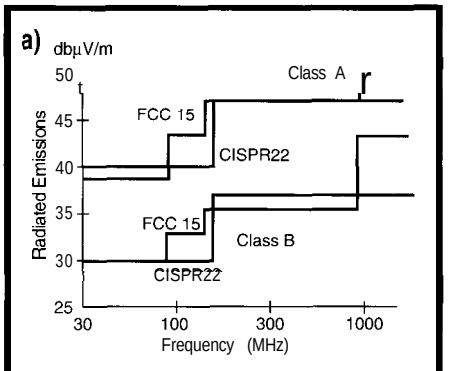
The FCC and the EU community limit the levels of radiated emissions a digital device can emit (see Figure 5). Note that the levels for Class B devices are lower than those for Class A.

Electronic equipment designed for home use is subject to more stringent levels. The test procedure consists of the EUT being placed at a distance from a receiving antenna.

It is operated, and the received radiation over the frequency band is recorded. If the reading is higher than the levels given in Figure 5b, the product fails.

The standard frequency range is 30-1000 MHz. However, as clock rates increase, the potential for undesirable emissions above 1000 MHz increases.

For this reason, the FCC requires testing up to 2 GHz if the highest frequency generated or used by the EUT ranges from 108 to 500 MHz. The requirement is up to 5 GHz if the highest frequency generated or used is in the 500-1000 MHz range. The Europeans check to the tenth harmonic of the highest generated or used frequency or to 1000 MHz, whichever is greater.



b)

Frequency (MHz)	FCC Part 15 Class A	FCC Part 15 Class B	CISPR 2 Class A	CISPR 2 Class B
@ 3 m				
30	49.5	40	50	40
88	54	43.5		
216	57	46		
230			57	47
960	60	54		
1000	60	54	57	47
@ 10 m				
30	39	29.5	40	30
88	43.5	33		
216	46	35.5		
230			47	37
960	49.5	43.5		
1000	49.5	43.5	47	37
@ 30 m				
30	29	19.5	30	20
88	33.5	23		
216	36	25.5		
230			37	27
960	39.5	33.5		
1000	39.5	33.5	37	27

Figure 5—*The graph (a) plots* the *radiated emissions /eve/s shown* in *the* fable *(b)* for FCC *15* and *CISPR* 22 *at 10 m. Levels* are rounded *to the* nearest *½dB.*

Figure 6 shows the test setup for radiated-emissions measurements. The tests are performed at an open field test site (OFTS). The boundary of the OFTS is an ellipse whose foci are formed by the location of the EUT and the measurement antenna.

In the area of the ellipse, objects capable of reflecting RF are prohibited. The common distances (d) are 3, 10, and 30 m. An inverse proportionality factor of 20 dB per decade allows comparison between 3-,10-, and 30-m distances (see Figure 5b). Therefore, if the limit is 30 dBμV/m at 30 m, it is 50 dBμV/m at 3 m.

Normally, an OFTS is in an area with a low background (i.e., a rural setting). The background at the OFTS must be at least 6 dB below the test limits.

The test standards are written to ensure that the test results are repeatable and that the emissions levels

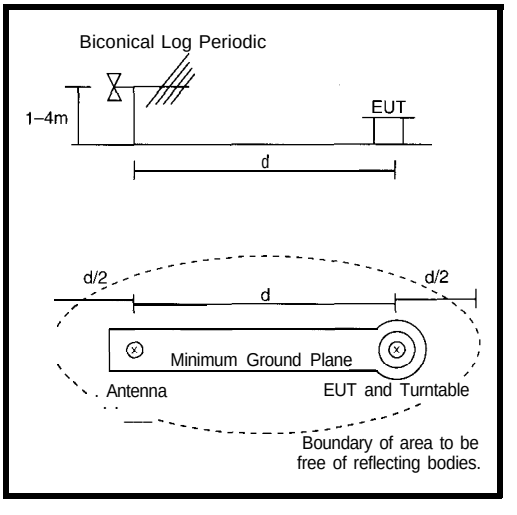


Figure 6—*The EUT is placed on a nonconduct-*ing fable *at a distance* (normally *3, 10, or 30 m) from the receive antenna. The antenna receives radiafed emissions from the EUT.*

detected are the maximum radiated levels. To ensure repeatability, a ground plane is used.

If no ground plane or only a partial ground plane is used, the results vary depending on ground conductivity, which varies with the seasons. Again, to ensure repeatability, the exact test configuration must be recorded and a photograph taken.

To ensure detection of maximum radiated levels, the EUT is placed on a nonconducting table. The table is placed on a turntable so it can be rotated fully through 360°.

The cables are moved to get maximum radiation. Also, it must be possible to raise and lower the receive antenna I-4 m and change the polarization between horizontal and vertical.

The EUT is run in the operating mode that produces the greatest emissions, even if it's not the normal operating mode. This requirement ensures that maximum radiation is detected.

To determine an EUT's radiated emission, there are several steps. First, you must select an appropriate frequency range to scan using the peak detector. Frequencies close to or above the limit are noted.

The frequency range scanned depends on the OFTS background levels. In areas with high background levels, the frequency scan must be reduced to make it easier to distinguish background from EUT radiation.

The peak detector is used because it's much faster than a quasipeak. The quasipeak detector takes 3 s/kHz of bandwidth.

For the second step, the frequencies identified are revisited one at a time. To maximize emissions levels, you must rotate the EUT, vary the antenna height, change the antenna polarization, move the cables, and change the operating modes.

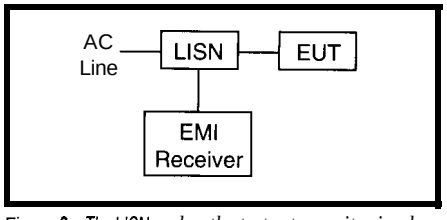
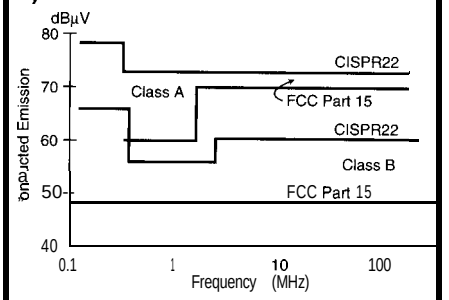


Figure 8—*The LISN makes the test setup quite simple. The LISN filters the incoming and outgoing AC line and couples the conducted emissions from the EUT to the EMI receiver.*

a)



b)

Frequency (MHz)	FCC Part 15 Class A	FCC Part 15 Class B	CISPR 22 Class A	CISPR 22 Class B
0.15			79	66
0.45	60	46		
0.50			73	56
1.71	**70**	**48**		
5.00			**73**	60
30.00	**70**	**48**	**73**	60

Figure 7a—*The plot of the conducted emissions levels for FCC 15 and CISPR 22 is taken from the values in the table (b). Note that the European limits start at 1.50 kHz. Levels are rounded to the nearest ½dB.*

When the maximum level for a specific frequency is found, the quasipeak detector is used for the final measurement. This step is repeated until all frequencies found during the first step are investigated.

Steps one and two are repeated until the full test range has been investigated [normally within 30-1000 MHz or higher).

CONDUCTED EMISSIONS

To keep the AC lines free from noise, the amount of noise that equipment can send back on the AC line is limited (see Figure 7). The basic test setup is shown in Figure 8.

The conducted-emissions test is far simpler than the radiated-emissions test. Like the radiated-emissions test, it must be repeatable, detecting the maximum emissions.

To determine the conducted emission from an EUT, you must follow a step-by-step procedure.

First, you should scan the frequency range 0.45030 MHz using the peak detector. Note the frequencies that are close to or above the limit. For European approval, the lower frequency limit begins at 150 kHz.

Then, one at a time, revisit the frequencies identified in step one. Maximize the emissions levels by moving the cables and, if applicable, altering the mode of operation.

When the maximum peak level for a specific frequency is found, the quasipeak detector is used for the final measurement.

If emissions at any one frequency are found to be over the limit using the quasipeak detector, then the average detector is used.

If the average detector's reading is 6 dB lower than the reading of the quasipeak detector, the quasipeak reading may be reduced (normally by 13 dB) before comparing to the limit.

This step is repeated until all frequencies found in step one are investigated.

MORE TO COME

Understanding these radiated- and conducted-emissions tests and the special equipment necessary to perform them will help you better design for EMI/EMC compliance.

But, this is only one side of the coin. Next month, I'll discuss the tests associated with immunity and susceptibility. ❑

Joe DiBartolomeo, ***I? Eng., has over 15 years' engineering experience. He currently works for Sensors and Software and also runs his own consulting company, Northern Engineering Associates. You may reach foe at*** *jdb.nea@* ***sympatico.ca or by telephone at*** *(905) 624-8909.*

REFERENCES

FCC, ***Code of Federal Regulations,*** Title 47, Parts 15 and 18, 1995.

IEC Standard 1000-4-1, ***Electromagnetic Compatibility, Testing and Measurement Techniques, Overview of Immunity Tests, Basic EMC Publication, 1992.***

C. Marshman, ***The Guide to the EMC Directive*** *89/336/EEC,* EPA Press, Ambo, UK, 1992.

T. Williams, ***EMC for Product Designers,*** Butterworth and Heinemann, Oxford, UK, 1996.

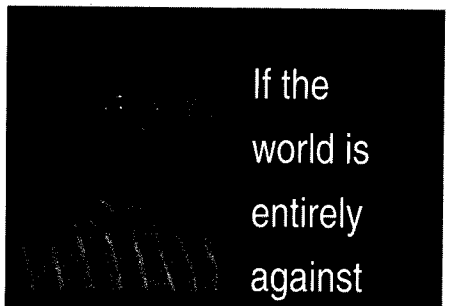
I R S

422 Very Useful
423 Moderately Useful
424 Not Useful

Magneto-Inductive Direction Finding

FROM THE BENCH

Jeff Bachiochi



If the world is entirely against you, then try the earth. Jeff shows how by using Earth's magnetic field with a Vector 2× compass module, you can use electronics to find your way out of the woods.

If you point your watch's hour hand at the sun, south is halfway between the sun and the 12 on the watch's face. Of course, this depends on a few things.

One, the sun is out. Two, you're in the northern hemisphere. And three, you don't own a digital watch.

Scouts are taught many ways to use the earth's resources to orient themselves in an unknown situation. But, when given a choice, they feel most comfortable using a magnetic compass.

Remember how our forefathers depended on the earth and their compass to guide them along an unfamiliar journey. Early oceanic explorers might have dropped off the edge of the earth if not for the magic needle.

But, the intriguing question—where does Earth's magnetic field come from?

To date, we don't have all the answers. The earth is like a dynamo, changing mechanical energy into electrical energy. With electrical energy comes magnetic energy.

The earth, however, is a homogenous dynamo. So, it's capable of creating currents without the use of wires. The turbulent motion within its electrically conductive liquid core generates a magnetic field.

The earth's dipole sets up field lines much like a bar magnet. Remember experimenting with iron filings in science. Sprinkling them on a piece of paper over a bar magnet indicated the invisible magnetic lines of force.

These same lines of force run between the earth's north and south magnetic poles. The farther you move from the earth, the weaker these magnetic lines of force become.

The magnetic compass (actually just a tiny bar magnet which rotates freely) reacts to the earth's magnetic lines of force. The south pole of the compass (usually painted red) is attracted to the earth's north magnetic pole, while the north pole of the compass is attracted toward the south magnetic pole.

The attraction of the two magnets (the earth's and the compass's) indicate the north/south we use for direction.

WHICH NORTH POLE?

You might think that since the earth spins on an imaginary axis extending through the north and south poles, its primary magnetic dipole would also extend through these points. It doesn't.

In fact, the magnetic north pole presently lies among Canada's Queen Elizabeth Islands as shown in Figure 1. If you place yourself between the Islands and the north geographic pole and walk in the direction your magnetic compass claims is north, you're actually walking south.

What's more, thanks to the flow of the earth's liquid core, the magnetic pole continuously moves around the north geographic pole at a rate of about 1 km per year.

Even though the magnetic north changes continuously, we can still navigate with extreme accuracy—thanks to magnetic declination. Magnetic declination tells us how far away true north (i.e., north geographic pole) is from where our compass points.

Every worthwhile map has a legend that defines the direction of true north. In addition, a declination constant is indicated so you can find true north by using a compass to find magnetic north.

Since this difference between magnetic and true north changes based on where you are on the earth, the declination constant might be wildly different on maps of different regions.

For instance, on the east coast of the U.S., add ~14° to magnetic north to get true north. On the west coast, subtract the magnetic declination since the magnetic north pole is currently between the two coasts (see Figure 2).

Of course, there are other factors which can throw off even finding mag-

netic north with a compass. Hard iron distortions abound. They may be man-made (e.g., a bridge, motor, or enclosure) or natural (e.g., an ore deposit). Any material that alters the earth's major dipole lines of force also alters the magnetic compass's movement.

DRIVING IN MY CAR

I've owned a Caravan since they were first introduced in the early '80s. The model I presently drive will pass 100k miles by the time you read this.

It features an overhead display of outside temperature and direction. Although direction is only broken down into eight bearings (N, E, S, W, NE, SE, SW, and NW), it's enough to indicate which way to go.

This device has often eliminated arguments about whether we're traveling in the right direction. While Beverly insists on checking with every passerby to ensure our progress is correct, I know exactly where we are at all times (except when I'm totally lost, and by that time, there isn't anyone around to get directions from).

The Royal Observatory in Greenwich, England, is the point on Earth where maps get their roots. The Greenwich Meridian is the imaginary line from north to south geographic poles and is 0" longitude.

Longitudinal lines run north and south on the globe, slicing it into 360 wedges, each equal to 1".

Each degree wedge can be divided into 60 wedge pieces, each 1 minute of arc. And, each minute wedge can be further subdivided into 60 slices, each 1 second of arc.

At the equator, a degree is 69.171 statute miles wide. This measurement gradually reduces to zero as you approach the poles.

So, 1 minute of arc is just over a mile at the equator, while 1 second of arc is -100'. You can see that, by indicating a longitude using degrees, minutes, and seconds, you can indicate any wedge around the earth to within 100'. All longitude lines are the same length from pole to pole.

What about locations within the wedge-to the north or south? Let's imagine we sliced the earth in half along the north/south axis. Now, place a protractor along this axis such that 90" is at north and 0" is at the equator.

Mark off on the edge of the earth the degrees between 0 and 90 and put the earth back together. If magic markers were suspended above a rotating Earth, the markers would draw rings around it.

These latitude lines have the same breakdown as longitude lines-degrees, minutes, and seconds. The equator is 0" and each of the poles are 90" north or south of the equator. Latitude lines differ in length from the longest at the equator to just a point at the poles.

ORIENTEERING

Thanks to cartographers, we can view areas on Earth with as much or as little detail as we wish. The newest format in maps is the 7.5 x 15 minute quadrangle. This map shows about 125 square miles of area.

The maps we used on a recent outing along the Appalachian Trail in

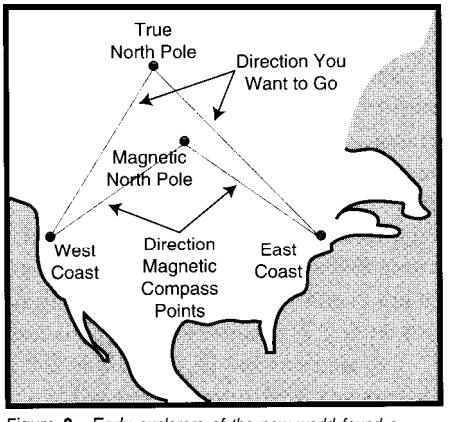


Figure 2—*Early explorers of the new world found a change in* declination as *they* traversed *from one coast to the other.*

Massachusetts gave a magnetic declination of 14" west of geographic north and a grid north of 1°5′ (i.e., 1 degree, 5 minutes) east of geographic north.

The map's longitude lines reflect the direction of geographical north, while the grid lines drawn every 1 km are on a grid north orientation. If we hadn't adjusted our course over the length of our IO-km hike by the declination given on the map, we would have missed our destination by a whole kilometer. Obviously, proficient orienteering skills are crucial.

Now, I admit you can't get much simpler than a magnetized needle free

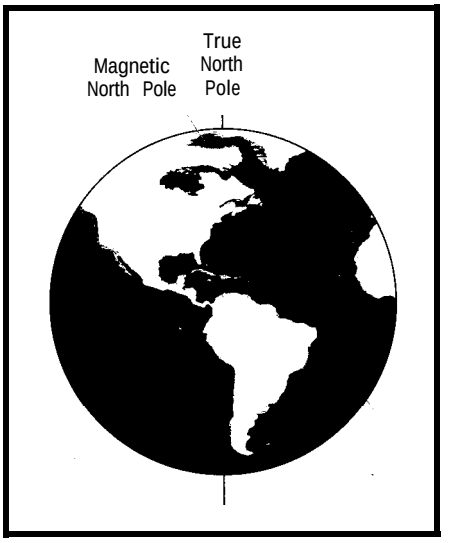


Figure 1—*I he magnetic north and true north are not the same. Thanks to Earth's liquid core, the magnetic* north moves about *1 km per year.*

spinning on an almost frictionless pivot. The compass is a wonder in its simplicity. There really isn't any room for improvement.

However, some of us are never satisfied to leave well enough alone. I've always wanted to make a digital compass with no moving parts. I figured I'd best get on with it before GPS gets so cheap it completely replaces the compass.

This month, I'll show you a low-cost two-axis compass module from Precision Navigation Inc. (PNI) which uses patented magneto-inductive magnetometer technology.

COMPASS MODULE

PNI uses a specially wound inductor as part of an oscillator circuit. The circuit's frequency is based on the coil's inductance.

If the coil is oriented so the earth's magnetic lines of force pass through the inductor's coils, they add to or subtract from the inductor's field as the circuit oscillates, changing the frequency of the oscillator.

If the circuit rotates (parallel with the ground) such that the coil moves in a circle (i.e., 360°), the frequency increases as the earth's field affects the coil in one direction. When the coil is perpendicular to the earth's lines of force, they have little or no effect on the frequency.

As the coil continues around (now facing the opposite direction), the

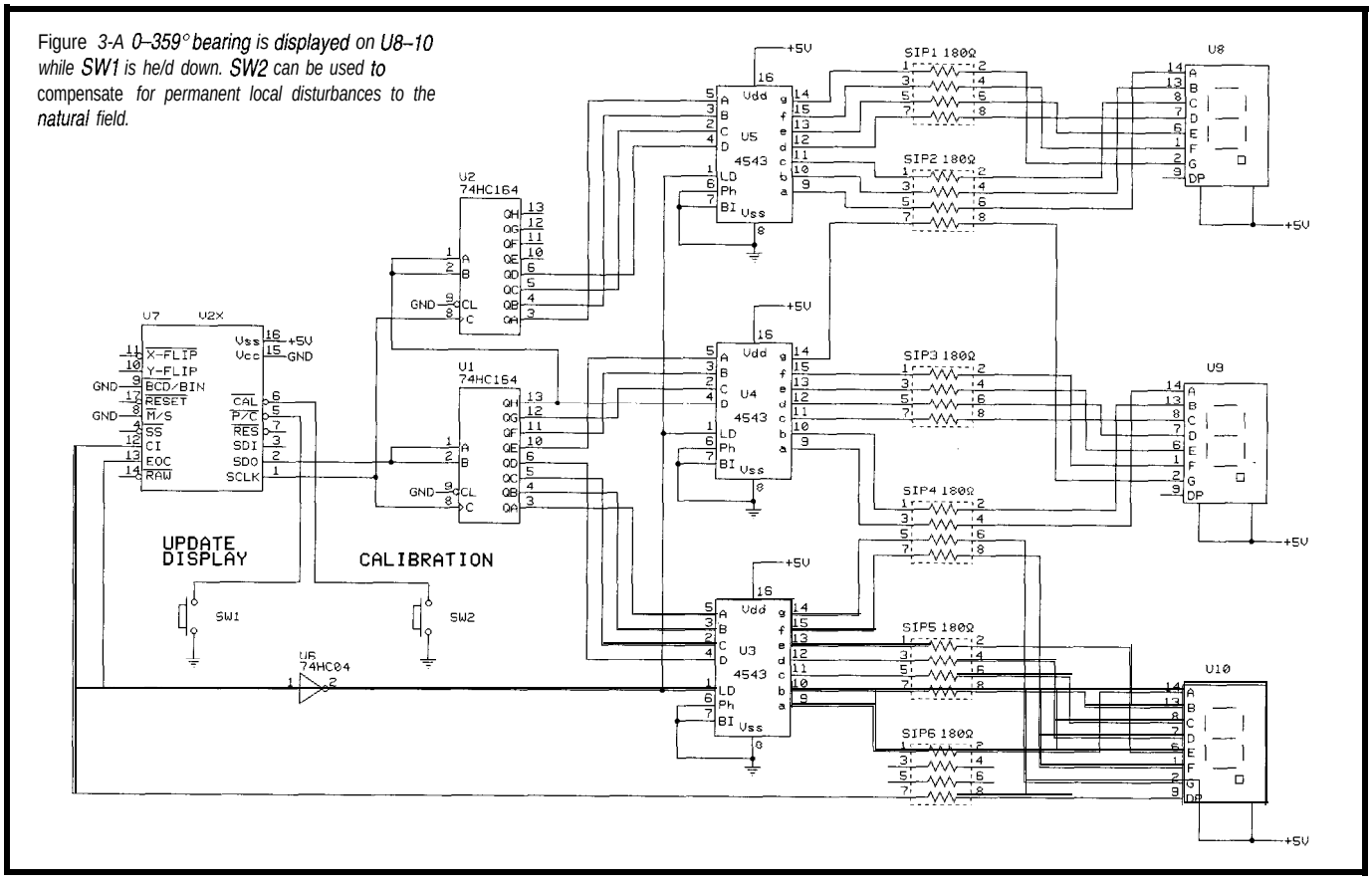


Figure 3-A 0–359° bearing is displayed on U8–10 while SW1 is he/d down. SW2 can be used to compensate for permanent local disturbances to the natural field.

frequency decreases as the lines of force now have the reverse effect on the circuit.

Still turning, the coil once again perpendicular receives little effect from the earth's field. Finally, the coil continues back to where it started with an increase in frequency.

Although we could get 360" of information from a single coil, the best accuracy occurs when the coil aligns with the earth's magnetic field. When the coil is perpendicular, accuracy is the poorest.

Directional accuracy can therefore be dramatically increased by using two coils at right angles to each other. Each coil covers alternating 90" areas.

PNI's Vector 2x compass module contains two such coils and the associated oscillator circuitry. In addition, a COB (chip onboard) processor evaluates the two oscillator frequencies and computes direction in degrees (o-359).

Clocked serial output can come as a binary or BCD number in a continuous or poled transmission. Because of the BCD output, you can connect the module to a display with little external circuitry as you see in Figure 3.

Set up for continuous serial BCD output, the 16-bit data packet can be transformed into four BCD digits by using a couple serial-to-parallel shift registers. The data format is most significant bits first with the upper six bits always being zeros. The upper digit is not even used as the maximum number since 359" needs only three places.

I chose the 4543 BCD latch decoder drivers so I could experiment with both LED and LCD displays. These drivers are very flexible. Just be aware that they can't supply gobs of current for superbright LED displays.

When you use them as LCD segment drivers, you must run an oscillator into the Ph(ase) input on the drivers to prevent permanently biasing the LCD in one direction (see my discussion of LCDs in *INK* 78).

CRUISE CONTROL

The 2x compass module has a handful of control inputs which increase its flexibility. Like most processors, the module has a *RESET input to restart the processor while the power remains applied. A minimum 10 ms low is required for reset.

The *M/S input lets you choose between a master mode (i.e., the module clocks out the data) and a slave mode (i.e., an external device may clock out the data). As shown in Figure 3, the master mode requires no additional processing power.

If the *M/S input is left high (i.e., slave mode], the external clocking source may choose slave mode with binary or BCD heading output data or raw binary output data.

With the *RAW input low, output data is increased to two 16-bit signed numbers. This raw data is not a heading but the data from the x and y coils used to calculate a heading.

To start a conversion, *P/C input is taken low for at least 10 ms. In master mode, a heading is clocked out the SDO by the internal SCLK when conversion is done (EOC output remains low during the conversion). In slave mode, the EOC rise signals the heading is ready and can be clocked out.

Dropping the *SS input low signals the module that you wish to clock out the heading. You now provide the SCLK to the module, and it outputs the heading via SDO.

Tying the ● RES (resolution) input low offers more precise calculations, although you get fewer conversions per second. The output heading normally increases as you travel clockwise from north at 0" toward east at 90".

Because the module can be oriented right side up or upside down, x and y coils need to be oriented correctly in relation to the north/south and east/west axes.

*XFLIP and YFLIP inputs reverse the orientation of the coils on each axis to allow for optimum positioning of the module.

As discussed earlier, certain materials can affect the path of Earth's magnetic lines of force. If the module was mounted near a metal mast on a sailboat, the mast would tend to warp the lines of force in the immediate area.

The Vector 2x module can compensate for permanent local disturbances by going through a calibration procedure. You can do this by lowering the 'CAL input for 10 ms (the CI output goes high indicating calibration mode), rotating the module 180°, and then lowering the *CAL input (10 ms) a second time.

The two readings taken internally should be opposite. Remove any disturbance by making an adjustment in future readings based on the difference in the calibration readings.

TINKER AWHILE

Now, if you wish to simplify the circuitry a bit, you could use a AY0438 (as in *INK 78)* along with an LCD segment display. Although the costs are roughly the same, wiring all these discrete components takes a while longer.

The advantage of the discrete circuitry is strictly one of experimentation, as LEDs or an LCD can be driven from the same circuitry.

Although this setup is sufficient for much compass work, there are instances when a more sophisticated rig is necessary.

Next month, I'll show how a microprocessor can add other functions to an electronic compass to make it a full-fledged orienteering instrument. I hope you'll all find your way back here next time. ❑

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

Vector 2x **Compass Module**
Precision Navigation Inc.
1235 Pear Ave., Ste. 111
Mountain View, CA 94043
(415) 962-8777
Fax: (415) 962-8776

Digi-Key Corp.
701 Brooks Ave. S
Thief River Falls, MN 56701-0677
(218) 681-6674
Fax: (218) 681-3380

I R S

425 Very Useful
426 Moderately Useful
427 Not Useful

MUSIC to Soothe the Savage Bits

SILICON UPDATE

Tom Cantrell

When you hear how this baby handles memory, it will be music to your ears. MUSIC's MU9C1480A LANCAM can split a 64-bit word, efficiently accessing both associative (CAM) and associated (RAM) data.

Many times, a part designed to solve one problem has unforeseen applications. One of the best examples is our friend the microprocessor. It was originally considered little more than an expedient way to implement a calculator.

Though not as dramatic, perhaps the MUSIC Semi MU9C1480A LANCAM is another example. As the name implies, the part targets a myriad of LAN gadgets (e.g., switches, routers, bridges, etc.) fueling the I-way frenzy.

Indeed, if you're involved in the digital comm biz, the LANCAM is a must see. I tend to side with those who fear the Internet is going to struggle under its own weight, so I welcome technologies that help my feeble little packets get where they're supposed to.

But even if your application has nothing to do with LANs, you may want to check out the LANCAM. It's an interesting chip that creative designers may be able to exploit in unique and unexpected ways.

RAM IN REVERSE

A CAM (Content Addressable Memory) is like regular memory in reverse. To use RAM, you give it an address and get access to the data. But with CAM, you give it the data and get back an address.

The concept of CAM isn't new. Indeed, you can find CAMs buried inside your PC involved with such tasks as virtual memory translation and cache control.

However, the '1480A is the first complete, fully integrated CAM chip I've ever seen. It looks simple enough from the outside (see Figure 1).

There's a 16-bit data bus (DQ0–15). A few control and status lines consume little more than half the PLCC-44 package pins. The rest is allocated to power, ground, and no connects.

While we're at a high altitude, it's easy enough to dispense with the basics of accessing the chip as shown in Figure 2. The interface is straightforward with *E (Enable) serving as the data strobe and *W (Write) determining the direction of transfer.

*CM (Command Mode) is essentially an address line that classifies a particular transaction as either command/status or data. The *RESET line is a clue that the '1480A is brainier than other memory chips.

Do note write data has to be valid by the leading edge of *E, so you might have to synthesize a late write strobe depending on your micro's timing.

Also, the detailed *E timing varies depending on whether the corresponding transaction is defined as short, medium, or long. For example, *E low pulse-width minimum (t_{ELEH}) for a 70-ns '1480A is specified as either 15, 35, or 55 ns, respectively. Hot-rod CPUs with the help of a PLD or ASIC may exploit the differential, but simpler designs can just treat every cycle as long.

The remaining half dozen or so pins reflect the unique function of the chip,

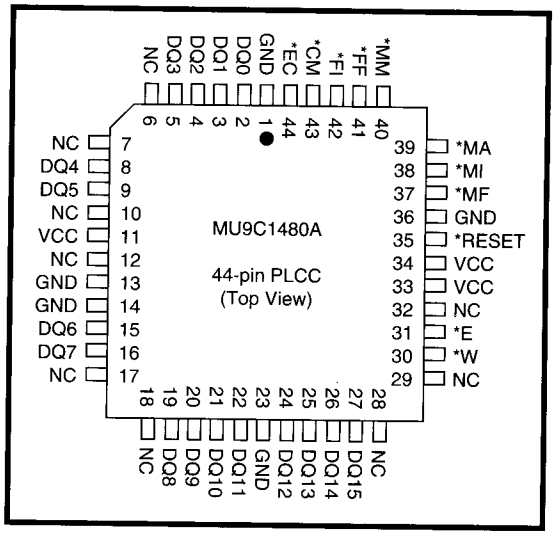


Figure 1—*The intelligence of the '1480A is hidden behind a simple 44-pin facade. The fact that there's no external address bus allows chips of different capacity (e.g., 1K × 64, 2K × 64, etc.) to plug into the same socket.*

so let's hold off on them till we get under the hood. For now, understanding how the '1480A works will be easier if we first consider the problem it's designed to solve.

HASHANDCACHE

Keeping track of all those little packets flitting hither and yon isn't easy. Ultimately, it boils down to managing tables and lists associated with addresses and other packet attributes.

Computers have been doing this kind of stuff a while. There's no shortage of literature describing the many suitable algorithms and data structures. A helpful way to make sense of them all is to consider the performance of typical operations (e.g., inserting, removing, finding, sorting, etc.).

The most obvious characteristics of any particular scheme are speed and storage efficiency, which exhibit an expected tradeoff. Basic constraints such as whether keys (e.g., LAN packet addresses) are unique and how sparse data is further guide your choice.

Subtle factors like symmetry can come into play. Many algorithms are optimized for quick reads at the expense of writes. The credit bureau wants to repeatedly access your record quickly, but it doesn't care much about how long it takes to create it (and, judging by horror stories, it cares even less about taking erroneous data out].

And then, there's determinism. To what degree is the timing of a particular transaction predictable and repeatable? For a credit-card transaction, exact timing doesn't matter as long as the checkout line keeps moving.

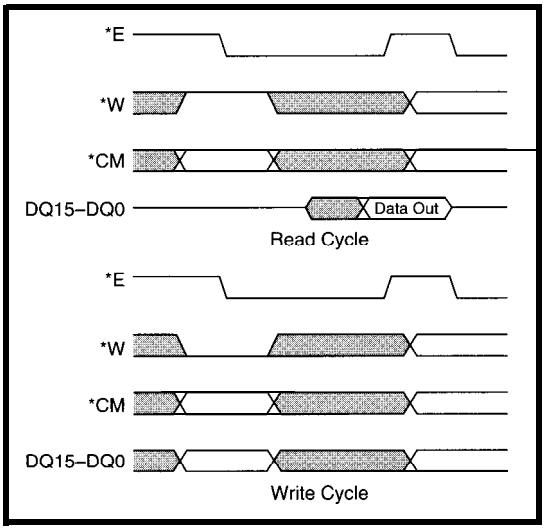


Figure 2—*The basic '1480A interface is a simple* three-wire (*E, ● W, and *CM) affair.

Even this superficial tutorial is enough to raise concerns about LAN-gear applications. With multichannels of data streaming in and out at many megabits per second, there's little time for complicated data structures and intricate access methods. A rather high degree of symmetry and determinism is preferable to avoid buffer-busting bottlenecks and pathological delays.

CAM t RAM ◻ CRAM?

Let's take a closer look inside the '1480A, which as you can see in Figure 3 has quite a lot of stuff. However, its primary functionality revolves around the 1K x 64 CAM array and the closely coupled 64-bit Comparand and Mask (MR1 and MR2) registers.

As mentioned, a basic CAM compares all 64 bits of the Comparand against the array and returns the address of a match, if there is any. The '1480A can do that, but it also offers the option of splitting the 64-bit word into CAM and RAM segments on 16-bit boundaries.

In other words, the 64-bit word contains both associative (CAM) and associated (RAM) data. For even more flexibility, the Mask registers can isolate particular bits for both CAM comparisons and RAM access.

In LAN applications, the CAM portion might contain a packet ID (i.e., a network address) and other descriptive info such as a time stamp, while the RAM portion holds a pointer to the packet's physical memory address.

More like a CPU than memory, the '1480A has an instruction set and registers used to set up and administer operation. Remember, from the outside, only four types of access are possible (read and writes with *CM high or low).

By default, command (● CM=O) writes go to the Instruction register, while command reads come from the Status register. The TCO (Temporary Command Override) instruction allows access to the other registers.

The TCO instruction specifies the register of interest which is accessed on the next command read / write cycle. Subsequent command cycles revert to the Instruction / Status default until another TCO is issued.

Similarly, data accesses (*CM=1) can be directed to or from the Comparand and Mask registers or the memory array itself by issuing the

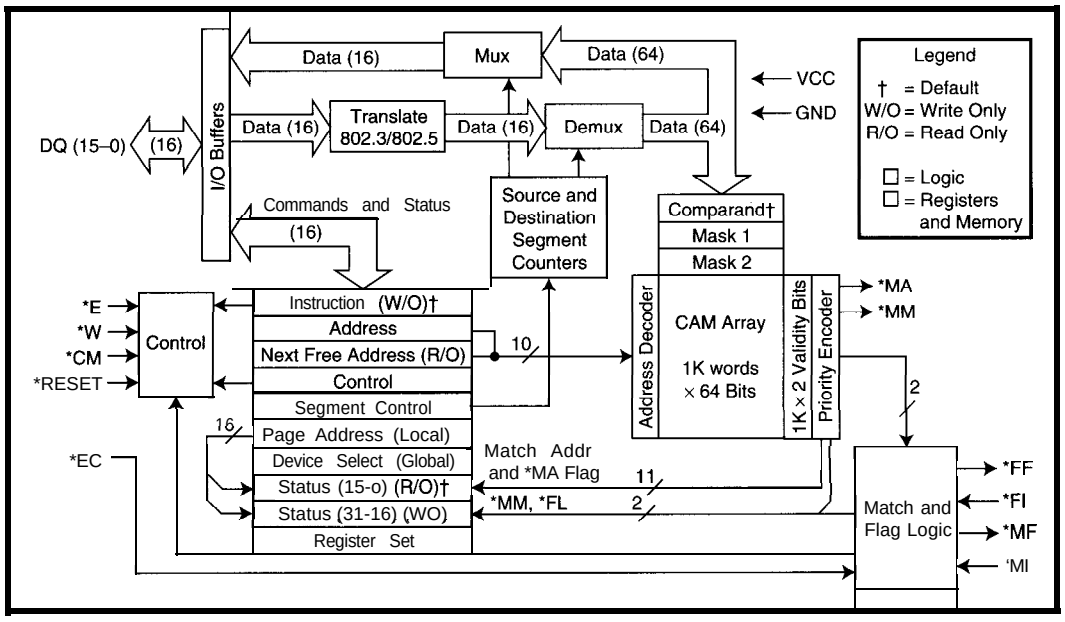


Figure 3-*A look* inside *the '1480A shows that the* chip *is more like a CPU than a memory.*

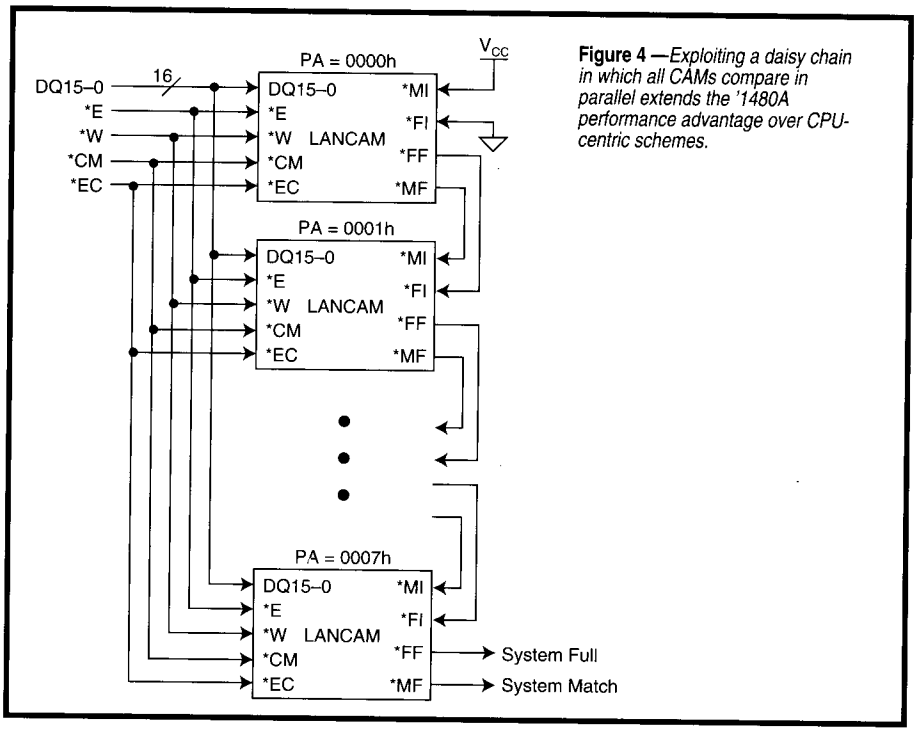


Figure 4—*Exploiting a daisy chain in which all CAMs compare in parallel extends the '1480A performance advantage over CPU-centric schemes.*

SPS (Select Persistent Source) and SPD (Select Persistent Destination) instructions. As their names imply, the chosen source and destination remain in effect until another selection is made.

Each 64-bit word in the CAM has two validity bits. These signify whether the corresponding entry is empty or valid or should be skipped for comparison. Validity settings can be checked with the CMP (Compare) instruction and manipulated with the VBC (Validity Bit Control) instruction.

Once everything is initialized, simply writing the Comparand register kicks the '1480A into high gear. That portion of the Comparand defined as CAM (and not masked with MR1 or MR2) is compared with all valid entries in parallel, returning a match status (via the Status register and *MA pin) in 20–30 ns (depending on speed grade).

Just for laughs, I wrote a program to linearly search a 1000-element array. It took about 2 ms to match up the 500th element (i.e., the average search length).

Admittedly, my old Mac isn't a speed demon, but since the CAM is about 100,000 times faster, I think you get the point. Yes, the chip does guzzle 1 W during comparison, but who worries about mileage at a roadrace?

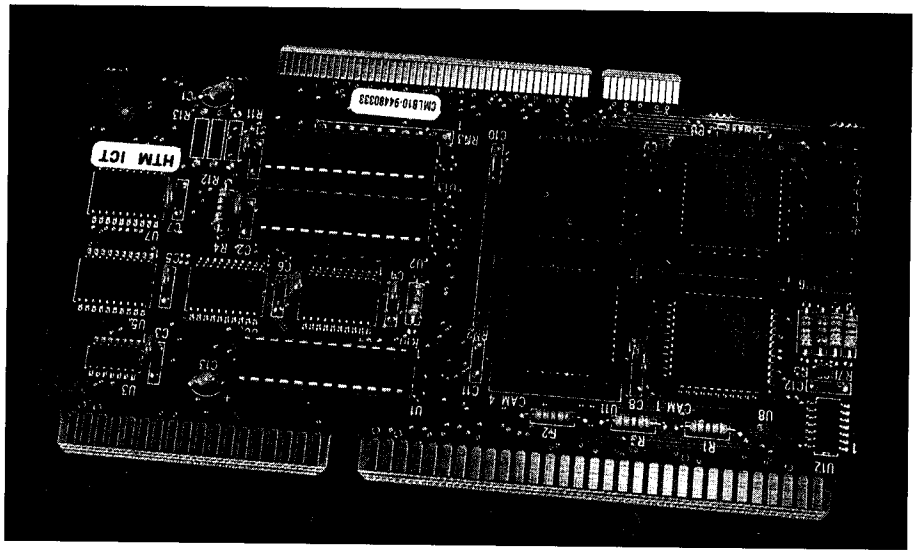


Photo 1—*Put a hot CAM under the hood of your PC with the $195 CAMLab.*

There's also a multiple-match notification (Status register and *MM pin), which is especially handy for servicing nonunique keys (e.g., more than one packet going to the same address) or conversely, signaling an error when multiple matches aren't allowed. It's useful in conjunction with CM P and VBC to tweak the validity bits at all matching locations in a single operation.

MOV can transfer data to or from the Highest Priority Match location (with multiple matches, it's the one with the lowest address). MOV can even direct data to the Next Free Address (validity bits show empty), which is good when you've got to dump data pronto.

Beyond embellishments such as dual-register sets (easing foreground/background operations), built-in shifter, and programmable address format (Ethernet/Token Ring), most of what's left involves a daisy-chain scheme for multiCAM expansion (see Figure 4).

The wiring is simple enough with each chip's *MF (Match Flag) and *FF (Full Flag) outputs connected to the next chip's *MI (Match In) and *FI (Full In) inputs. *EC (Enable Chain) allows temporarily disabling the chain for initialization and housekeeping.

In a multichip setup, the last ● MF and 'FF outputs reflect the status for the entire chain. Overall performance is only limited by flag in/out prop delay, which is less than 10 ns per device.

CAM CAN DO

It's not hard to see how the '1480A accelerates LAN-gear applications. The chip blows the doors off CPU-driven sort-and-search schemes, and the $20 toll is all it takes to bypass I-way gridlock. MUSIC also offers larger LANCAMs and, soon, smaller ones as well, with a forthcoming 256 x 64 version projected to cost less than $10.

If you want to play, check out CAMLab ($195). This PC plug-in board (see Photo 1) combines up to four '1480As with all the software you need.

Other applications can benefit, too. MUSIC app notes describe a number of possibilities such as fully associative disk cache, accelerated data compression, and RTOS task-list management.

Perhaps a fuzzy accelerator, database engine, or interpreting machine sparks your interest. Creative designers may find that a little imagination—and a CAM or two-goes a long way. ❑

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 by fax at (510) 657-5441.

SOURCE

MU9C1480A LANCAM, CAMLab
MUSIC Semiconductors
254-B Mountain Ave.
Hackettstown, NJ 07840
(908) 979-1010
Fax: (908) 979-1035
http://www.music.com/

I R S

428 Very Useful
429 Moderately Useful
430 Not Useful

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**

Well, I never thought a column based on electronic communications could become obsolete, but that's just what's about to happen.

We started the Circuit Cellar BBS over 10 years ago as a way for Circuit Cellar readers to exchange information and retrieve article-related software. It's been (and continues to be) a unique gathering p/ace for some of the brightest minds in our industry to shoot the breeze about any technical issue that happens to come up.

ConnecTime began in the first issue of Circuit Cellar INK *as a way to spread the good news about what was happening on the BBS.* Many *of the* on-line discussions were just too good to keep to a *small group of callers, so I shared them with our readers. Since calling the BBS direct/y was too expensive for many, this column became a way for them to share a bit of the on-line world each month.*

The Circuit Cellar BBS has gone through several transformations over the years, and the most exciting is about to happen. With a switch *to new hardware and software, the BBS and all its files will be free/y available to anyone with Internet access. You'll still be able to call the modem lines and use a familiar character-based interface. However,* **you'll** *also have the option of using telnet to access the messages, ftp* to access the file area, or your *favorite* Web *browser* to access both *the messages and files graphically.*

So, why does this change make ConnecTime obsolete? Given that the vast majority of our readers have at /east some Internet access and would therefore now have free access to the BBS, we felt the *pages* of the *magazine* could *be* better *used for other types of editorial. Rest assured that while you'll no* longer *be seeing* what's *going on on the BBS in these pages, that activity is still there, Join us and have some fun.*

I do want to thank a few individuals who have contributedabove and beyond the call over the past 10 years (and, I'm hopeful, for years to come). Specifically, Bob Paddock, Pellervo Kaskinen, George Novacek, Jim Meyer, and Ed Nisley. I could *always* count *on one of these* sharp guys to *have an* intelligent, well-thought-out answer to some of *the most off-the-wall questions I've ever seen. Thanks also go to all our callers, for without them, there wouldn't be a Circuit Cellar BBS nor would there have been a ConnecTime.*

Assuming all goes well in the coming weeks during the upgrade, I'll have a more detailed article next month describing how the new sys*tem works and how you can access it. See you on the net.*

EPROM Life Span

Msg#: 3194
From: Gordon Brandly To: All Users

I'm in the process of checking over my old Osborne 1 and IMSAI computers and making them run again-hey, someday they're gonna be worth something again, right?

I remember hearing or reading somewhere that data stored in EPROMs of that vintage had about a lo-year life span. Since the data is stored as electrical charges on little "floating" MOSFET gates, the charge is bound to leak away over time. I'm going to read the contents of these EPROMs with my programmer and store them away on my main hard drive, since it gets backed up regularly. My question is, has anyone here experienced EPROMs losing/corrupting their contents due to age? I'm wondering if I should be reprogramming these precious EPROMs of mine. I hate to touch them more than I have to-the less I mess with them, the less opportunity for Murphy's Law to be invoked.

BTW, I'm going to rewrite all of my floppies, too. I've heard that floppy data also has a lifetime of about 10 years or so. I wonder if anyone has done a formal test of that?

Msg#: 3241
From: Pellervo Kaskinen To: Gordon Brandly

I believe it was early last year that I described my personal findings on the life of the data in an EPROM. It appears that the manufacturers have done plenty of testing and then wanted to play it safe. The lo-year endurance quote is very conservative.

In fact, I tried to erase three misprogrammed OTP chips (actually microcomputers with internal ROM). Elevated temperature should accelerate the discharge rate. So, I baked my chips for several days and nights to the effect of nearly 100 years life. Not a bit changed! Granted, these are newer chips with more accumulated knowledge in the manufacturers' bag of tricks, but still....

Msg#: 3291
From: Gordon Brandly To: Pellervo Kaskinen

Thanks for the information, Pellervo. That would make me breathe easier, if it weren't for some new findings:

I've now read three of my Osborne EPROMs (two boot ROMs and one character generator]. Each has a sticker with a 4-digit hex number on it. Luckily, I noticed that this number matched the checksum my programmer displays when it reads a device. One of the boot ROMs matched perfectly, but the other has a very different checksum (and of course that's the one I normally have inside my Osborne)!

The character-generator ROM's checksum was off, so I whipped up a program to display its contents in character-cell form. It appeared that at least one bit had changed from 1 to 0, and another two bits changed from 0 to 1. I'm not sure what's going on in that first case. I assume that "old" bits would *only* change from 0 to 1, since the erased state of this type of EPROM is all 1s.

CONNECTIME

Msg#: 3418
From: Pellervo Kaskinen To: Gordon Brandly

The internal structure of an EPROM contains more than just the memory cell (capacitor). There is the sense transistor or multiplexer arrangement. And, there are a number of possible leakage locations as well as chances for "bad solder joints"-actually the aluminum interconnects. So, both an empty cell or a lost contact or leakage path can result.

In the late '70s and early '80s, plenty of attention was put on the alpha particle effects that might discharge the EPROM cells or might cause "soft errors" on DRAMs. That was even believed to set a limit on the reduction of the feature size and the maximum memory size.

The semiconductor industry found that the alpha particles came mostly from certain isotopes in the process contamination. After that, it was a rush to improve the process, building much better clean rooms and so on. And the problem has been kept under control. And the same actually applies to the other possible failures I mentioned. The connections are better, and the transistor geometries and mask alignment reduce the risk of developing leaks or poor parameters on the multiplexing matrix.

So, as you probably have some of the old generation chips, you might indeed want to copy them over on newer technology chips. 2716, eh? Or were they even bigger? A friend of mine did have his Osborne open several times, but it's too many moons ago. Moreover, it could have been a different version anyway. Anyway, it looks like a couple of vendors are continuing to supply even the small chips. National has been our source lately for the few 2716-type chips we need. Actually, it is 27C16, and we cannot get the original slow 350 ns (or 450 ns) chips. 200 and 250 may be also gone as you read this. All of this indicates smaller feature sizes and potentially shorter storage, but I think that is not actually a danger to worry about.

The most committed source *appears* to be SGS-Thomson. Their products are sold at least by Mouser and either Jameco or JDR; I don't remember which one for sure.

Msg#: 3450
From: Gordon Brandly To: Pellervo Kaskinen

Thanks for that interesting and detailed explanation. The Osborne 1 uses a 2716 for its video character generator and a 2732 for its boot code. I'm hoping to scrounge a 27C32 for that boot ROM, since (as you mentioned) it's a more modern and reliable part. I hope that someone out there has a verifiably good revision 1.44 ROM that they can lend me (or a file of its contents). My 1.44 ROM might or might not be OK. Without another ROM, it's going to be very difficult to tell. My Osborne boots fine, but I'm worried that some software is going to call some ROM routine where one crucial bit got changed for the worse!

As for sources, I'm sure of Jameco at least. I just received my first Jameco catalog in about ten years. They're still selling a lot of the same chips they had in my old catalog! :-) I couldn't believe my eyes-1 thought production stopped on most of those old timers years ago!

Anyway, thanks again for your help.

Msg#: 3380
From: Steve Ciarcia To: Rufus Smith

Since one of the comments on this thread has to do with one-time programmable EPROM, have any of you guys heard of ways to erase them? I once heard that they could be reset with x-rays. No doubt, there is a whole chat group on the Internet working on this subject ;-)

Msg#: 3393
From: Chuck Olson To: Steve Ciarcia

Jim Meyer (who apparently doesn't get on this BBS anymore) told me that he erases his OTP parts with x-rays. Of course, he has access to a particle accelerator at his job.

Jim stated that fairly intense x-rays are required-the medical machines aren't of any help. A machine like they use to x-ray welds at a nuclear power plant should work.

This might be a business opportunity for Hare Krishnas (they could put OTP-filled luggage through the airport baggage x-ray machines over and over until erasure).

Msg#: 3475
From: Jan Verhoeven To: Steve Ciarcia

My Roentgen tube died last month. Seems I need to get another one.

I think you could very well be right, but which person (except for Pellervo, who might have x-ray equipment from his welding lab) has access to x-rays? It would be a very safe calculated risk to keep in this option of erasing.

Msg#: 3526
From: Pellervo Kaskinen To: Steve Ciarcia

We had a discussion of this topic about 12 to 18 months ago. The only one with positive results was Jim Meyer, who had used a research cyclotron for his source of x-rays. That source was, however, down for several months due to remodeling. Whether it was back up by the time of Jim's last contact was never established.

Theoretically, *anything* more energetic than the 253.7-nm wavelength of the UV light should do. It is only a question of the dose *and* its interaction with the target silicon. It is possible that some of the radiation passes right through the chip just like the bulk of neutrinos pass through the entire globe.

There has been a long tendency of reducing the potentially harmful doses of medical x-rays. That may have in-

CONNECTIME

creased the required exposure times to a level of being impractical, but I still claim that a dental x-ray machine can erase most of the OTP chips, given enough time or repeating the exposure a number of times. There is also a danger of damaging the nonmemory areas of the chip with wrong wavelength radiation, but I do not know how real that danger is. A prudent application would naturally be using minimal doses, with testing between the exposures. Once all cells are "empty," give an additional dose of some 10–20% of what has accumulated by then. Or, test the chips with both minimum and maximum supply voltage. They are deemed erased when neither minimum nor maximum operating voltage detects a programmed cell. No overdose is required.

Msg#: 3807
From: Pellervo Kaskinen To: Jan Verhoeven

No, we do not have or desire to have any radioactive material or x-ray guns around. Just like in the European countries I'm familiar with, here also the burden of qualifying and requalifying everything is more than we want to be dealing with.

Also, for metallurgical testing, we would need "harder" (i.e., shorter wavelength) x-rays than what most medical devices use. That would also cause some side effects, like crystal damage on silicon, if the radiation stops in the silicon at all. Mostly that kind of radiation just passes through. It would not achieve the erasing of the OTP chips we're talking about here.

It is my understanding that the medical devices ***have*** to be more productive-maximum charge elimination in the memory cells with minimum harm to the crystal structure. I think Jim Meyer argued against this theory, but I'm still not sure what to believe.

Regarding x-rays and welding, here is a bit more for those interested in miscellany. There are two occasions of x-rays in the welding environment. Probably 99% of them are in the inspection of seam integrity. You put a strip of photographic film along the weld seam and a radiation source on the opposite side of the part. The film is exposed by the radiation and, after ordinary development, it reveals any cracks or inclusions in the seam because those areas do not attenuate the x-rays as the solid metal does.

The second occurrence of x-rays in welding is inadvertent. It comes when Electron Beam Welding generates the radiation exactly the same way as the famous Roentgen tube does. It does not help the welding-indeed everybody would be happy if this side effect could be eliminated. But at 100-kV acceleration, when the electrons hit ***any*** metal, x-rays are one of the results. Of course, the Roentgen tubes like to use mostly durable metals, such as tungsten, as the target, while in EB welding the anode material is whatever has to be joined.

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.

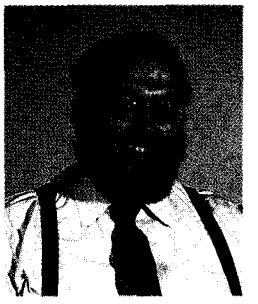
ARTICLE SOFTWARE

Software for the articles in this and past issues of ***Circuit Cellar INK*** may be downloaded free of charge. Currently, files are on the Circuit Cellar BBS. In the future, the files will also be 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.

431 Very Useful 432 Moderately Useful 433 Not Useful

PRIORITY INTERRUPT

Beware of Complacent Convenience



As many of you know, the subject of home automation and building control is near and dear to my heart. Unfortunately, it's also a technical province that has little general understanding and even less applied expertise. When you mention "home control" to the average person, they picture automatic lawn mowers and robots that deliver mixed drinks. Even among the technically literate, home control is typically something that only the "gearhead engineer" next door would install or use.

Over the years, Ken and I have tried to present home control as a problem whose solution is a critical combination of strategic building blocks. As an embedded-control-oriented magazine, the design and application of these blocks fit naturally within *INK*'s primary mission-technical enrichment and real application. People who've used our HCS II realize that the same hardware could control an industrial process just as easily as monitoring a home HVAC system. The knowledge gained for one applies to the other.

The one factor that inhibits universal application of home controllers, ours included, is that they require the user to define a control technique and detail the task sequence. In short, you have to install and program it.

For someone who designs or uses embedded control systems as a profession, this is easy. The thousands of HCS II owners among our readership are just applying what they do every day. And, while it's anything but convenient to string wires through a hot attic, these "gearheads" know why it works, how it works, and when it works.

Sometime in the past, I might have predicted that "home control" could reach acceptance on its own merits. Today, I believe that the installation of "convenience technology" will ultimately supplant the need for a dedicated HCS installation. Given the potential performance of future PCs, expanded communication channels, and intelligent appliances, "home control" will become a by-product of a lifestyle centered on interactive convenience.

Recently, we had an article on using the Internet as a vehicle for communicating with an embedded system. How long before someone offers an Internet-connected black box that goes on your furnace or HVAC and allows your furnace repair company to tailor optimum energy consumption, automatic fuel delivery, load shedding, and billing? Sounds pretty convenient.

In another room, the combined PC/TV, document scanner, video digitizer, DBS receiver, fax, and messaging center forwards your mail and calls, watches your driveway for visitors, orders some preselected catalog items, downloads a movie for later, orders a pizza, checks the kids' homework, schedules the evening TV viewing, and adjusts the surround sound according to room occupancy. Even more convenient?

I could go on and on. Super PCs, HDTV, and ISDN are the means. The acceptance of all of this stuff will be in the name of convenience. Unfortunately, I also see a very real downside to this much connectivity. Is a world that screams about Orwellian oversight and "big brother" ready to give up this much individuality in the name of convenience? The natural extension of all these convenience services is the compilation and dissemination of user profiles with the most intimate details.

Today's HCS takes time and effort to employ, but it's still a private undertaking. Tomorrow's HCS may simply be a subset of an all-encompassing "convenience." Before we concede that fate, however, we should give more thought to complacent acceptance without realizing the consequences of all this electronic connectivity. Beware of the wolf in sheep's clothing.

steve.ciarcia@circellar.com