CIRCUIT CELLAR INK®

THE COMPUTER APPLICATIONS JOURNAL

#79 FEBRUARY 1997

EMBEDDED PC SPECIAL BONUS SECTION: EMBEDDED PC

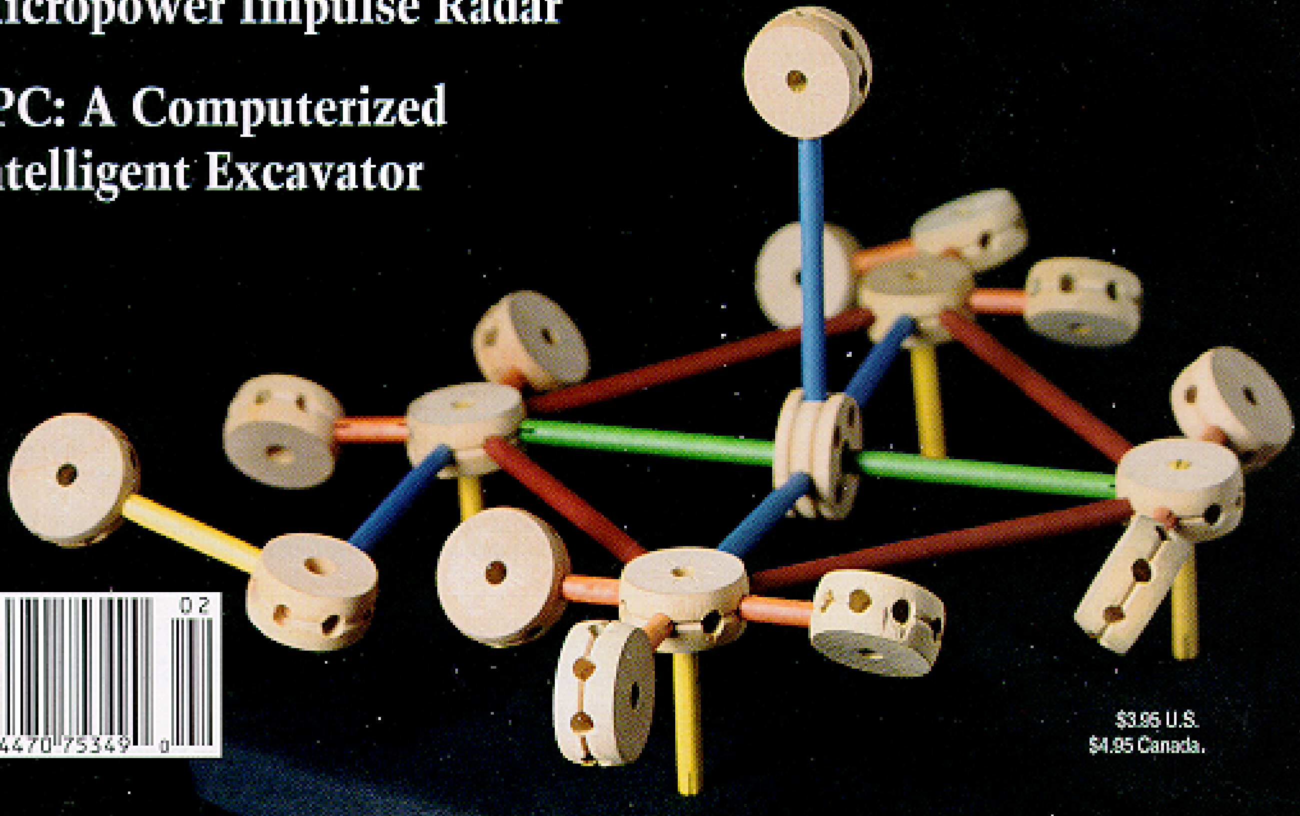
DISTRIBUTED CONTROL

BACnet: An Object-Oriented Network Protocol

Embedded TCP/IP

Micropower Impulse Radar

EPC: A Computerized Intelligent Excavator



$3.95 U.S.
$4.95 Canada.

TASK MANAGER

Gloom and Doom'?

It's so obvious, I can't pass it up. When you think 'distributed control, you think network. And these days, when you think network, you think Internet.

I've been involved with the Internet at various levels for about three years now, though it seems like a lot longer given the developments we've seen in that time. The good I've seen has been very good: faster connections, easier to use E-mail, and cheap, powerful Web browsers. The bad has been very bad: junk E-mail, out-of-control newsgroups, negative media attention, and streaming data.

While a lot of doomsday naysayers have predicted the impending failure of the Internet, I don't hold those feelings. I think we simply have to be careful about how we use it.

Consider the power grid and the phone system. While both are generally reliable and meet the needs of the many, they do fail-sometimes due to nature, sometimes human error, and sometimes overdemand. Whatever the cause, we take it in stride. Nobody screams that the entire power grid or phone system is going to hell in a handbasket.

Think back to the last time there was a brief large-scale outage of Internet service. Popular press headlined doom and gloom for the overburdened, under-designed Internet. But, has any of this come to pass? Not the last time I checked. Certainly, things slow down in the afternoon and evening when everybody is taxing the system, but power companies also threaten rotating blackouts when everybody's air conditioners run simultaneously on a summer afternoon. I still routinely experience the pleasure of a quick and almost instantaneous response from a Web server thousands of miles away first thing in the morning and on weekends.

Bursty traffic from E-mail and Web usage won't be the biggest problem on the Internet. It will be streaming data. The network just wasn't designed for lots of continuous data required by real-time audio and video. When I want to listen to a talk show or watch a sporting event, give me radio and television. Unless we keep the Internet to textual and small graphical data, we'll certainly face many of the problems Steve anticipates without significant investment (see Priority Interrupt, p. 104).

The issue at Hand

Our first feature article gives a networking primer, covering some of the basics of any connected setup. Next, we look at adding the language of the Internet—TCP/IP—to embedded control. I find the idea of connecting controllers to the Internet for worldwide instant access particularly exciting. We then turn to BACnet, another network standard, that bases communications on data content rather than target addresses. Finally, we take a look at some of the latest in DC-to-DC converters.

In our columns this month, we start a new MicroSeries concentrating on design issues related to EMI, Jeff decodes BASIC-52 tokens, and Tom contemplates the possibility of personal radar.

In this month's *EPC*, we look at a sample application where off-the-shelf embedded controllers and some wireless communication combine to control a water-treatment plant. Next, we continue our look at the embedded Internet with another off-the-shelf networking application. In this PC/104 Quarter, we present one of the winners of our "wanna be famous?" design contest. And in Applied PCs, a desktop PC is networked with an embedded PC without much difficulty at all.

editor@circellar.com

CIRCUIT CELLAR INK®

THE COMPUTER APPLICATIONS JOURNAL

FOUNDER/EDITORIAL DIRECTOR
Steve Ciarcia

EDITOR-IN-CHIEF
Ken Davidson

MANAGING EDITOR
Janice Marinelli

TECHNICAL EDITOR
Elizabeth Laurençot

ENGINEERING STAFF
Jeff Bachiochi

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:
Pellervo Kaskinen

PUBLISHER
Daniel Rodrigues

ASSOCIATE PUBLISHER
Sue Hodge

CIRCULATION MANAGER
Rose Mansella

CIRCULATION ASSISTANT
Barbara Maleski

CIRCULATION CONSULTANT
John S. Treworgy

BUSINESS MANAGER
Jeannette Walters

ADVERTISING COORDINATOR
Dan Gorsky

CIRCUIT CELLAR INK®, THE COMPUTER APPLICATIONS JOURNAL (ISSN 0896-8985) is published monthly by Circuit Cellar Incorporated. 4 Park Street. Suite 20, Vernon, CT 06066 (860) 875-2751. Periodical rates paid al Vernon, CT and additional offices. One-year (12 issues) subscription rate U.S.A. and possessions $21.95, Canada/Mexico $31.95, all other countries $49.95. All subscription orders payable in U.S. funds only. via international postal money order or checkdrawn on U.S. bank.

Direct subscription orders and subscription related questions to Circuit Cellar INK Subscriptions, P.O. Box 698, Holmes, PA 19043-9613 or call (800) 269-6301.

POSTMASTER: Please send address changes to Circuit Cellar INK, Circulation Dept., P.O. Box 698, Holmes, PA 19043-9613.

Cover photography by Barbara Swenson
PRINTED IN THE UNITED STATES

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

HAJAR ASSOCIATES NATIONAL ADVERTISING REPRESENTATIVES

NORTHEAST & MID-ATLANTIC	MIDWEST & SOUTHEAST	WEST COAST
Barbara Best	Christa Collins	Barbara Jones & Shelley Rainey
(561) 694-2044	(954) 966-3939	(714) 540-3554
Fax: (561) 694-2051	Fax: (954) 965-6457	Fax: (714) 540-7103
B.Best-Hajar@worldnet.att.net	HajarChrista@worldnet.att.net	shelley.hajar@worldnet.att.net

Circuit Cellar BBS—24 Hrs. 300/1200/2400/9600/14.4k bps, 8 bits, no parity, 1 stop bit, (860) 871-1988; 2400/9600 bps Courier HST, (660) 871-0549. World Wide Web: http://www.circellar.com/

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

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

Entire contents copyright © 1997 by Circuit Cellar Incorporated. All rights reserved. Circuit Cellar INK is a registered trademark of Circuit Cellar Inc. Reproduction of this publication in whole or in part without written consent from Circuit Cellar Inc. is prohibited.

EMBEDDED PC

See Pages 49–80 for our Special Bonus Section

INSIDE ISSUE 79

READER I/O

I'M NOT CONVERTED

I'm eternally frustrated with bigger, badder, faster computers doing less and less. I used to play PacMan on Ataris, TRaSh-80s, and Apple IIs. Now, Microsoft says I need a '486DX2/66 with 8 MB of RAM to do the same thing. So, when I read Steve's editorial ("Software-The Real Generation Gap," *INK* 74), I applauded loudly.

But, when I turned to the front and read Ken's column ("The Windows 95 Convert"), I almost suffered a coronary. It seemed Ken was bragging about the same thing Steve was preaching against.

I've tried to convince fellow software developers that Windows has a lot of advantages as a development environment. Having used Windows 3.1 for developing software for both DOS and Windows, I agree with all of the advantages that Ken mentioned for Win95. But, it's obvious that Ken has never used Win3.x for this purpose.

Let me explain my Windows 95 upgrade experience. The wallpaper painting is now a visible process. My hard drive is almost continually grinding. Starting Notepad (a 34-KB app) is an exercise is patience. Printing-once painless-now slows down my word processor.

I think *INK* is the best source of information in a world driven by media hype. But, Ken, your column read like a Microsoft ad and has tainted the image I once held of your publication. I'm tired of the same hardware doing less and less. You know that the capability of any one piece of hardware does not degrade with time. So, why is my '386DX-40 having difficulty with simple word processing, when I used to be able to do it on a 2-MHz Z80?

Jon Foster
jafco@teleport.com

I sound like a Microsoft ad and it's tainted your view of INK!!

Perhaps I'd accept that accusation if Microsoft owned Circuit Cellar INK. *However, neither I nor the magazine have anything to gain by singing the praises of their products. My commentary is just that-opinion.*

Indeed, Windows 95 has not given me the troubles it has you. It's as I said, "snappy." I never had the ease of multitasking with Windows 3.1 that I have with Windows 95. In fact, I was continually so frustrated with Windows 3.1 that I avoided it like the plague. All my useful work was done under DOS.

So, let's suffice it to say we have different opinions. Certainly, INK's *reputation extends beyond a single half-page editorial opinion-or at least it ought to.*

Ken Davidson

TAKE TWO

I enjoyed reading "Video Timecode Fundamentals" by Ingo Cyliax *(INK 77).* I commend him for tackling such a misunderstood subject. Unfortunately, the article contains a couple errors.

The first and most glaring mistake concerns SMPTE, which is the Society of Motion Picture and Television (not "Tone") Engineers. Second, the location of the VITC signals in the video signal is inaccurate.

The ABC defines the position of VITC as being on line 14 (field 1) and line 277 (field 2), and TOD appears on line 18 (field 1). It appears that the line numbering is off by a count of 7. NTSC video-line counts start at the first pre-equalizing pulse. First comes the pre-equalizing pulse sequence (i.e., 2 equalizing pulses per line, or 6 equalizing pulses over 3 lines). After the vertical serration sequence (i.e., 2 serration pulses per line, or 6 serration pulses over 3 lines) comes the post-equalizing pulse sequence, which is the same as the pre-equalizing pulse sequence.

These take a total of 9 lines. No signals are allowed on line 10. Lines 11-21 (field 1) and 274-284 (field 2) are used for data and test signals. That's the general makeup of the VBI. For a quick introduction to SMPTE time codes, check out <http://www.sfoundry.com/pages/tech/smpte.htm>.

Tom Michener
tmichenr@erols.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 off ices at (860) 875-2199.

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/**.

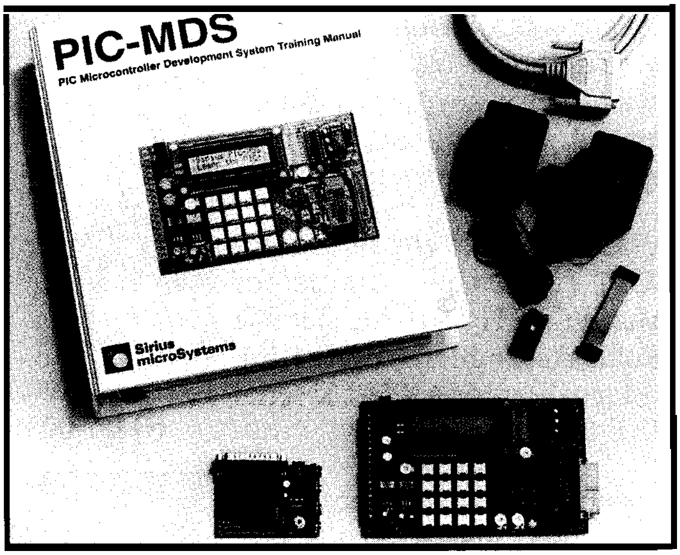
NEW PRODUCT NEWS

Edited by Harv Weiner

PIC DEVELOPMENT SYSTEM

The PIC-MDS-a complete development system consisting of a development board, programmer, cross-assembler, and detailed training text-was specifically designed to simplify microcontroller programming and use. Step-by-step examples guide you from writing and testing simple programs to keypad, LCD, A/D, serial I/O, data-logging, and interrupt routines.

The PIC-MDS is ideal for learning how to design in microcontrollers. Since commonly used microcontroller I/O devices are included, you can refine applications on the development system without hardware prototyping. Once applications are proven, you can transfer the programmed PIC to the target system.



You can create and assemble microcontroller applications on a PC using the PM Macro Assembler or you can write your own applications incorporating the tested software routines. Applications are programmed into the PIC micros via the included EPIC programmer. The EPIC programmer programs all midrange 18-pin PICs and uses optional 28- and 40-pin socket adapters for larger devices.

The PIC-MDS development board includes both PIC-16C71 and PIC16C84 microcontrollers, 16-key matrix keypad, 2-line x 16-character LCD display, ZIF socket, buffered-LED-port state indicators, and an RS-232 serial port. Also included are two analog input potentiometers, 256-byte serial EEPROM, oscillator socket for crystals or ceramic resonators, an onboard +5-V and variable DC power supply, and screw terminal and header connections to all PIC I/O pins.

The complete PIC-MDS development system sells for $299.

Sirius microSystems
172 Harvard Rd.
Waterloo, ON
Canada N2J 3V3
(519) 886-4462
Fax: (519) 886-4253
http://www.siriusmicro.com/

#500

KEYBOARD-CONTROLLER FAMILY

A highly integrated keyboard-controller family that exceeds stringent industry-standard requirements without external components has been announced by Zilog. The **Z86Kxx** series improves electrostatic discharge (ESD) and electrical fast transient (EFT) performance, and it reduces electromagnetic interference (EMI).

The new chips are capable of withstanding in excess of ±20 kV system-level ESD with no hardware or software failure. When transient signals are injected into the PC power cord and propagated through the system, the device withstands ±2 kV of EFT. The distributed clock edge, combined with Zilog's proprietary PADs, results in a 10-dB reduction in EMI.

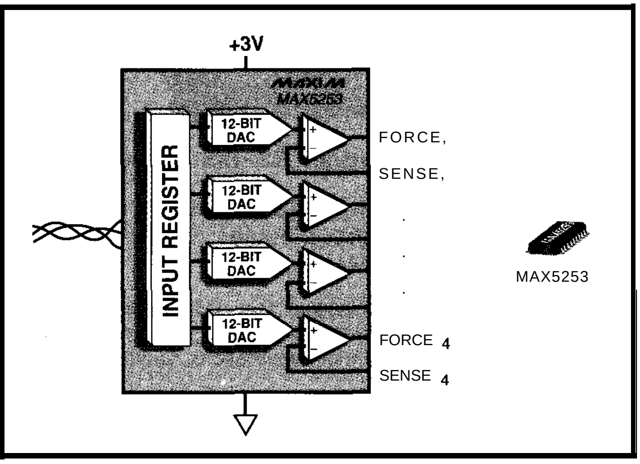
Six different devices (Z86K13–Z86K18) optimize ROM sizes, packages, and clock sources. They have 200 bytes of RAM and are in 40-pin DIP, 44-pin PLCC, and 44-pin QFP packages. ROM varies from 2 to 4 KB. You can also choose between an RC oscillator or resonator.

The devices are priced under **$1** in high volume.

Zilog
210 E. Hacienda Ave.
Campbell, CA 95008
(408) 370-8000
Fax: (408) 370-8056

#501

NEW PRODUCT NEWS



QUAD D/A CONVERTER

The **MAX5253** is a monolithic, quad, 12-bit DAC that operates on a single 3.0-3.6-V supply. It contains double-buffered input registers, four 12-bit DACs, four precision output amplifiers plus control logic, and a serial interface. Accessible feedback connections for the output amplifier (force and sense pins) enable remote sensing, specific gain configurations, and high-output driver capability. Each amplifier provides rail-to-rail output swings.

Other features include a low-power shutdown mode that lowers the 0.9-mA quiescent current to 20 µA maximum and an internal, power-on reset that guarantees all outputs are 0 V when power is applied. Also included is a user-programmable, general-purpose logic output for the serial control of other external devices.

The MAX5253 has a 3-wire interface compatible with the SPI/QSPI and Microwire synchronous-serial standards. The DAC registers can be updated independently or simultaneously, and all logic inputs are TTL/CMOS compatible. Prices start at **$11.35** in 1000-piece quantities.

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

#502

Good, Fast, Low Cost:

PICK 2

With PromICE, You CAN Have It All!

High performance memory emulation and debugging:

- Stable and reliable on today's embedded systems.
- New faster access speeds now standard.
- The best connection solutions for **TSOP**, PSOP and **PLCC** chips.
- Expanded Virtual UART support for industry-standard debuggers.

Ultra-Fast code downloads reduce development time:

- New high-speed download support for Windows NT
- 90 **KBytes/Second** over a PC parallel port.
- tow-cost Ethernet support for UNIX systems.

New lower Prices for 1997:

- 128 **KByte PromICE** now just $495.
- Source-level debugging systems at a fraction of an ICE's cost.

Today's PromICE really is "better, faster, less expensive". Call 1.800.PROMICE (776-6423) for more information, or visit us at www.gei.com.

Voice: (614) 899-7878 • Fax: (614) 899-7888 • email: info@gei.com
921 Eastwind Drive • Suite 122 • Westerville, Ohio 43081 • USA
US and Canadian Sales: 1-8OO-PROMICE (776-6423)

#107

NEW PRODUCT NEWS

LINEAR OPTOCOUPLERS

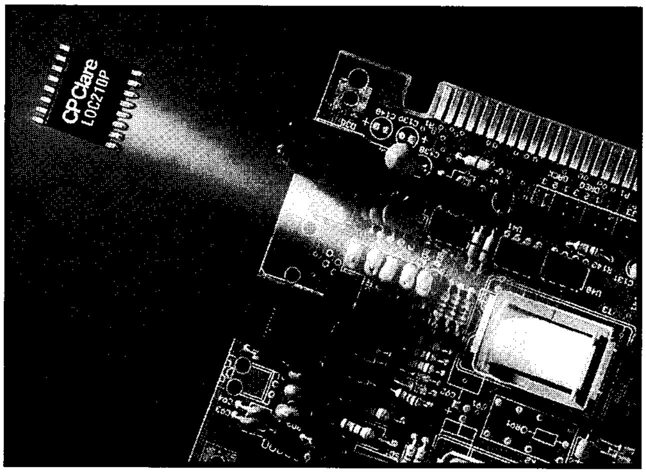
Two new dual linear optocoupler models tailored for digital telephone isolation, power-supply feedback voltage sensors, and medical-sensor isolation have been introduced by CP Clare. The **LOC210P** and **LOC211P** are also ideal for isolating process transducers, audio signal interfacing, and modem transformer replacement.

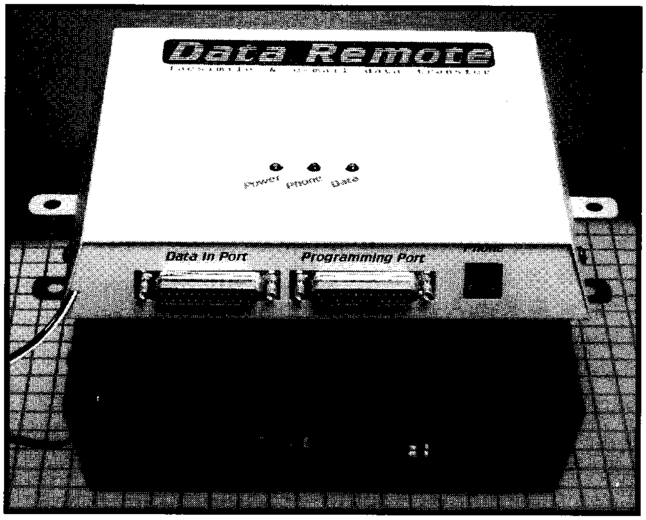
The LOC210P and LOC211P each provide two independent linear optocouplers in 16-pin SOIC packages. Each optocoupler features an infrared LED optically coupled with two phototransistors. One input phototransistor generates the servo control signal that compensates for the LED's nonlinear time and temperature characteristics. The second phototransistor provides an output signal that's linear with respect to the servo LED current. The compensated optocouplers achieve a better than 0.01% servo linearity and greater than 200-kHz bandwidth. Both optocouplers also feature high gain stability and low power consumption. Offering better than 87-dB THD and 3750-V_{rms} I/O isolation, the devices replace the transformer, providing a low-profile integrated solution.

The LOC210P sells for $2.74, and the LOC211P sells for $3.31 in quantity.

CP Clare Corp.
North American Sales Off ice
601 B Campus Dr.
Arlington Heights, IL 60004
(847) 797-7000
Fax: (847) **797-7023**
http://www.cpclare.com/

#503





REMOTE DATA REPORTER

Data Remote attaches to data loggers, chart recorders, network servers, phone systems, computers, or any device with an RS-232 output port to give instant reporting capability over standard phone lines. It constantly receives a stream of ASCII text information from the attached equipment and stores it internally. Using standard line power and a simple phone line connection, Data Remote sends reports to fax machines, remote computers/modems, or Internet E-mail addresses through Phonetic's Internet server.

Data Remote has two alarm functions-an inactivity alarm and alarm keyword detection. Activity through the serial port is monitored, and a period of inactivity generates an alarm message, which can be crucial when there's nonfunctional equipment.

The presence of up to 20 specific keywords is also monitored, and when encountered, Data Remote immediately delivers an alarm message to up to 32 user-programmable destinations.

In addition to fax, computer, and E-mail, Data Remote can send alarm messages via alphanumeric or numeric pagers.

Data Remote is priced under **$1000.**

Phonetics, Inc.
901 **Tryens** Rd.
Aston, PA 19014
(610) 558-2700
Fax: (610) **558-0222**

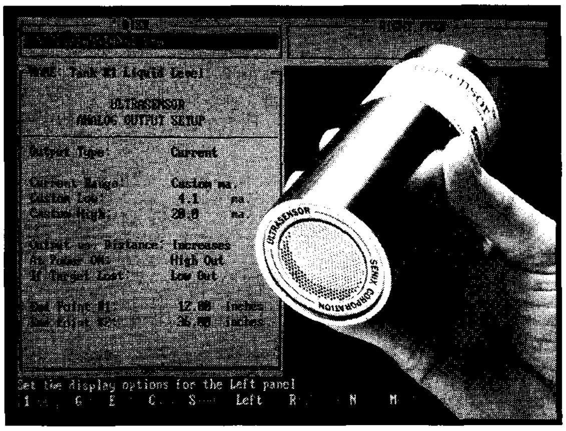
#504

NEW PRODUCT NEWS

CONFIGURABLE MATERIAL SENSOR

The Ultra-U ultrasonic sensor offers noncontact distance measurement of materials for positioning, proximity, liquid/solid levels, dimensioning, and motion control. Measured distance varies from 2″ to 37′, with a maximum resolution of 0.004" and a repeatability of 0.1% of distance. The sensor's outputs are proportional to or set by the measured target distance.

This updated model features a linear 4–20-mA current-sourcing output in addition to the 4–20-mA current-sinking output. With both current-loop interfaces, the sensor easily connects with any 4–20-mA current-loop device, whether single-ended or differential input. These devices include PLCs, recorders, data-acquisition systems, motor drives, and displays.



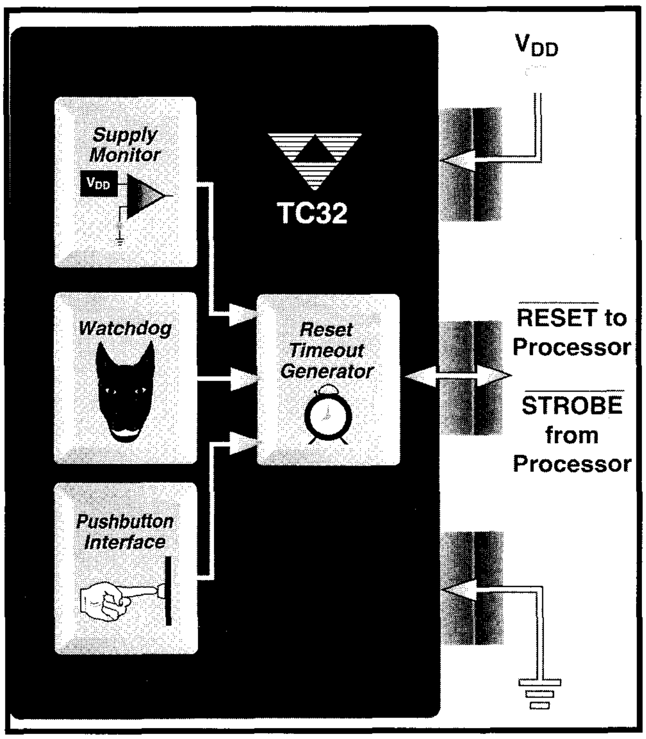
The Ultra-U's operating parameters can be configured using Senix's SoftSpan program running on a PC-compatible computer. You can set the measurement period, ignore unwanted objects, control outputs, and adjust range characteristics to suit specific applications. A rear push button adjusts many parameters without a computer. Parameters are permanently stored in the sensor and easily duplicated.

Ultra-U is packaged in an HIPS plastic enclosure. The Ultra-U-SS in a stainless-steel case is also available. Both models include interfaces such as 0–10-VDC voltage, 4–20-mA current loop (sourcing and sinking), two transistor switches, and serial data RS-232 in one package.

Senix Corp.
52 Maple St. • Bristol, VT 05443
(802) 453-5522
Fax: (802) 453-2549 **#505**

PROCESSOR SUPERVISOR

The **TC32M** is a fully integrated processor supervisor in a three-pin package. It safeguards processor sanity via three important functions-precision power-on/off-reset control, watchdog timer, and external reset override. The chip is well-suited for applications needing a fully integrated processor supervisor (e.g., automotive instruments, battery-powered computers and controllers, embedded controllers, and critical processor power monitoring).



The TC32M maintains the processor in the reset state whenever supply voltage is out of tolerance. Once the supply is within tolerance, the TC32M maintains the processor in reset for an additional 250 ms, allowing time for system stabilization. During normal system operation, the processor must apply pulses to the TC32M at a prescribed rate. Should the processor fail to do this, a system fault is indicated and the TC32M issues a momentary processor reset. The TC32M's output may be wire-ORed with an external auxiliary reset signal or a push-button switch for reset-override control. When connected to a push-button switch, the chip provides contact debounce.

The TC32M is available with operating temperature ranges of 0 to +70°C or -40 to +85°C. Pricing in an SOT-223 package is $0.68 and $0.72, respectively. In a TO-92 package, costs are $0.60 and $0.65, respectively.

TelCom Semiconductor, Inc.
1300 Terra **Bella** Ave. • Mountain View, CA 94043-1836
(415) 968-9241 • Fax: (415) 967-1590 **#506**

FEATURES

FEATURE ARTICLE

Bill Payne

A Networking Primer

Networking is essential in today's computing environment. Bill takes the time to bring us all up to speed on the basic components of networking. Here, he focuses on networking evolution and concepts.

As technology evolves, networking and internetworking are becoming more prominent in our daily lives. Both Microsoft and Novell a more products, leading us further into a networked world.

Novell intends to bring networking into some of the less-conspicuous areas of our lives through smart utility meters and Internet-attached TVs at home. In light of this rapidly changing technology, it's important that hardware and software engineers understand how the underly of networking interoperate.

In this three-part series, I cover all the basic components of networking. In Part 1, I discuss the concepts behind networking and its evolution. Next, I explore media types, topologies, and protocols. And, in the final installment, I delve into the internetworking devices.

So, fasten your virtua let's begin our journey.

The concept of networking was driven by the need to s tion and hardware devices. As programs became smarter, they needed more resources than a single machine could offer.

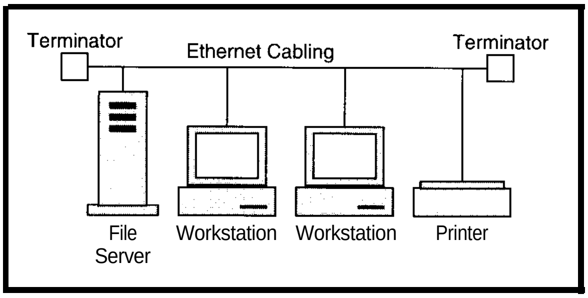


Figure 1—A standard Ethernet LAN using coaxial cable comprises one file server, two workstations, and a LAN-attached printer. The terminators at the ends of the coaxial cable ensure proper operation of the LAN.

In the early '80s, the prevalent networks consisted of multiple dumb terminals communicating to a common host computer and sharing a printer. These early networks were based on serial communication technologies like RS-232, RS-422, and even 4–20-mA current loops. Both the UNIX and IBM worlds used devices referred to as terminal multiplexors to support multiple clients on one host.

With the introduction of the personal computer, the idea of a centralized intelligence came into question. Rogue groups of programmers started emerging who did not fit into the highly structured world of the mainframe.

As the personal computer matured, users developed a need to share data and printers. These networks-known as Sneaker Nets-were typified by the image of programmers in tennis shoes carrying diskettes between systems. These nets worked well for basic data transfer.

The high cost of printers was truly the first driving force behind building a Local Area Network (LAN). To this day, networks are primarily used for file and print sharing.

Novell is a good example of a file- and print-sharing network OS. Microsoft's network OS functions as an application server similar to a UNIX system supporting true client/server applications.

As Figure 1 illustrates, a basic LAN comprises a file server, one or more workstations, a printer, and a physical cabling scheme to interconnect the various devices. I'll discuss the devices that interconnect LANs into inter- and intra-networks (e.g., routers, gateways, and bridges) in Part 3.

FILE SERVER

The component—the file server-is generally the most powerful computing device in the network. It runs a true 32-bit multitasking OS to support multiple users and functions. It requires a significant amount of DRAM, compared to a workstation running Windows.

The file server supports multiple users via high-speed network interface drivers and boards. It uses a highly efficient disk I/O subsystem for handling issues like high-speed access and fault tolerance.

Hard drives within the file server are usually configured as separate volumes, which are further subdivided into directories. This arrangement is similar to the subdirectory structure used by DOS-based systems.

All shared network software applications are stored in this structure. Network users can also store common data files here.

The various volumes defined on the file server may not be accessible to all users. Different security levels can be established for volumes, directories, and even files, allowing or disallowing access to each user. In the file server, each user is a separate task with its own resource needs.

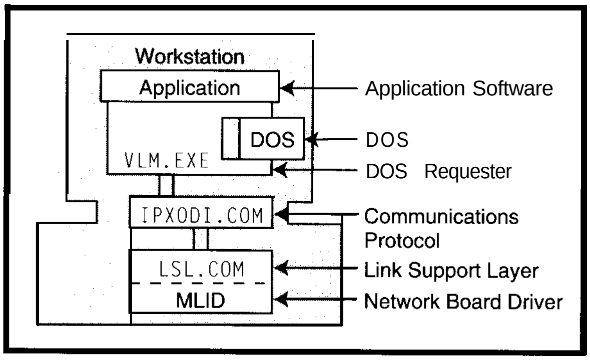


Figure 2—This model shows how the different pieces fit together inside a LAN-attached workstation. The DOS Requester (VLM.EXE) is shown as wrapping around the DOS module. It isolates the workstation operating system from the application software and the network.

In a simple network, printing functions are also handled by the file server. It establishes print queues, which are accessible to all network-attached users.

When a user directs an application's output to a specific network printer, the print data goes into a print queue on the file server. The network OS periodically queries all print queues to see if anything is in them. If print data is present, the network OS begins streaming data to the specific printer associated with that queue.

WORKSTATIONS

In this article, I define a workstation as a stand-alone DOS computer. Given my experience and training, I'll bias my treatment of this subject matter to the way each component works in a Novell environment.

In any network, each workstation must have a network board and con-

Listing 1—This Novell NET.CFG file resides on a LAN-attached workstation. The link driver, EXP16ODI, is provided by the manufacturer of the network interface card (NIC). This card is configured to support four different protocol stacks. The first drive letter available to the network is defined as drive F.

```
link driver EXP16ODI
  int 3
  port 300
  frame ETHERNET_802.2
  frame ETHERNET_802.3
  frame ETHERNET-II
  frame ETHERNET-SNAP

Link support
    BUFFERS 8 4202
    MEMPOOL 4096
  Max Boards 4

NetWare DOS Requester
    FIRST NETWORK DRIVE = F
    PREFERRED SERVER = CPI_Tul_1
    NAME CONTEXT = "OU=TULSA.O=CPI"
```

nection software installed. The workstation DOS provides the basic local services for file-system, screen-display, printer, and communications access to the serial ports.

DOS Requester connects an application to the network services as shown in Figure 2. DOS Requester wraps itself around the workstation's DOS.

All DOS requests generated by an application program must pass through DOS Requester. Similarly, all actions generated by the workstation DOS must also pass through it.

In this way, DOS Requester binds the application and DOS to the network services. In the Novell world, DOS Requester is loaded into the workstation memory by VLM.EXE.

COMMUNICATIONS PROTOCOL

A communications protocol is a set of rules used by two devices to move data across a network. It's the equivalent of a language spoken between two people. If neither party understands the other's language (i.e., rules), no data transfer is possible. If both parties use the same rules, they are guaranteed to communicate.

Novell uses the Internetwork Packet exchange (IPX) protocol to communicate between devices on a network. The Open Datalink Interface (ODI) specification allows multiple protocols on the same network cabling.

Each protocol on the workstation is referred to as a stack, which is not to be confused with the hardware definition of a stack. A protocol stack is the set of rules for that specific protocol.

For example, both Novell IPX and TCP/IP can be run on the same workstation using the same network board. Novell IPX uses one stack, and TCP/IP uses another.

The ODI specifications permit up to four protocol stacks to be attached to each network board. Any network board designed to meet the ODI specifications can communicate with any other vendor's network board.

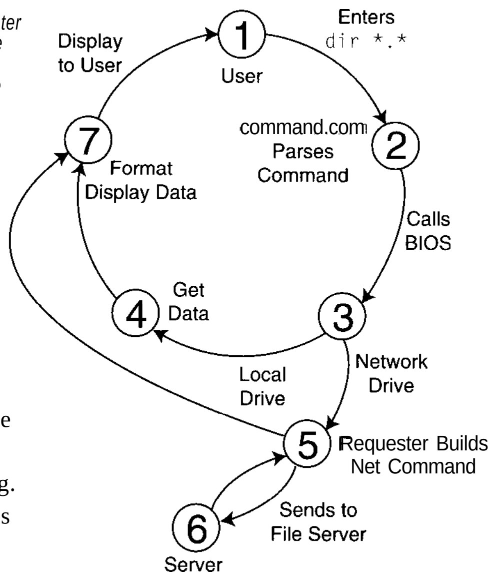


Figure 3—*This diagram details the logic flow which occurs when a user enters dir *.*. DOS Requester takes over at step 3 and determines whether the drive is local or network attached. The final result is displayed to the user as if both paths were local to the workstation.*

The Link Support Layer (LSL) acts as a switchboard between the communications protocol and the network board driver. It routes the information received from the physical network connection to the appropriate protocol stack and vice versa.

MLID

Novell's Multiple Link Interface Driver (MLID) is a special type of network board driver. This module understands and controls the network board at the hardware level. It connects the workstation's software at the LSL level with the physical network connecting to the device.

Novell uses IPXODI.COM to configure the various options for each network board in the workstation. This program reads the text data contained in the NET.CFG file. An example of a NET.CFG file is shown in Listing 1.

The network board driver is the Intel EXP16ODI driver. This 16-bit driver complies with the ODI specifications. The interrupt and port address are specific to this type of network board. The driver is provided by the network-board manufacturer.

The four lines beginning with "frame" are the protocol stacks for this board. It can communicate using the Ethernet 802.2, 802.3, II, and Sub Net-

#109

work Access Protocol (SNAP). I'll define these in detail in Part 2.

The Link Support section defines the memory pool's size in kilobytes on the network board. It also defines the number of communications buffers on the network board and their size in bits.

The NetWare DOS Requester section defines the logical drive letter used for the first network drive. This logical drive is where the user finds the login program.

OPERATIONAL DESCRIPTION

I'll now delve deeper into how DOS Requester operates. First, let's look at what occurs when a user enters a basic directory command at the workstation console. Then, I'll show the events involved in printing from an application running on the workstation.

The sequence of events that occurs when a workstation needs access to a network drive is depicted in Figure 3. The user at the workstation enters dir f:*.* at the prompt (assuming the f drive is a network drive). The command interpreter at the workstation parses the command and calls a specific BIOS system routine.

The BIOS system routine in ROM is not executed. Instead, the command is redirected to DOS Requester. The Requester checks whether the command requires a local-system or a network drive.

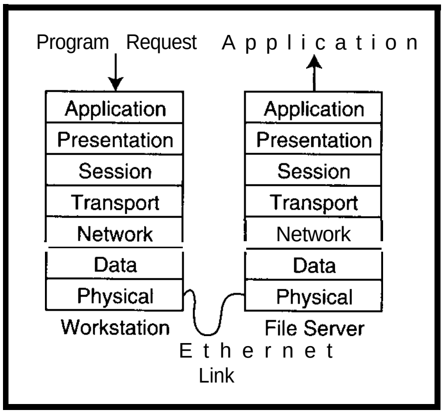


Figure 4-A program request such as a file command flows from the workstation up to the application program residing on the file server. The request is passed through the seven layers of the protocol model before being transmitted on the interconnecting media. It is then shipped, layer by layer, by the receiving seven-layer protocol model and finally passed as the original data to the application program.

If the command uses a local drive, it's passed on to the BIOS system routine. But, if the command uses a network drive, a network command is built by the Requester and sent to the file server.

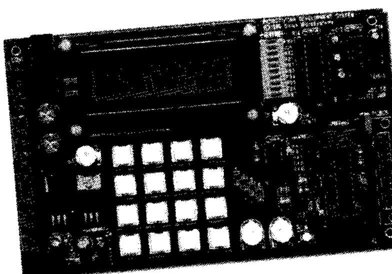
The file server processes the command and returns the requested data to the workstation. Finally, the data displays to the user at the workstation as if it had been accessed on a local drive.

The mechanics of the network message follow the guidelines defined by the international OSI model. This model is based on a seven-layer protocol used by both the workstation and file server to encapsulate and reconstruct the command (see Figure 4).

Network printing uses the same basic DOS Requester mechanisms as the file-system command parser. The user at the workstation can redirect any local printer port to a network-attached printer.

In a Novell environment, this function is accomplished via capture. The variables used in capture tell DOS Requester when to bypass the internal

Your First PIC!



The PIC-MDS is a complete training and development environment for the mid-range PIC™ microcontrollers. Proven in classrooms, the PIC-MDS lets you learn to use and develop applications for PIC microcontrollers quickly and easily. Our step-by-step examples guide you from simple programs through to keypad scanning, serial I/O, LCD display, A/D conversion, data logging, and interrupt routines.

The PIC-MDS Professional Pack includes:

- PIC16C71 & PIC16C84 microcontrollers
- EPIC Programmer & PM assembler
- detailed training manual with disk
- in-circuit programming cable & socket
- ZIF socket • buffered LED port indicators
- 2 X 16 LCD display • 2 analog pots • RS-232 port
- 256 byte serial EEPROM • crystal/resonator socket
- 5V & variable DC power supply • AC voltage adapter
- all I/O pins on screw terminals & PICBUS connector
- all code examples in Microchip and Parallax™ syntax

only $299!

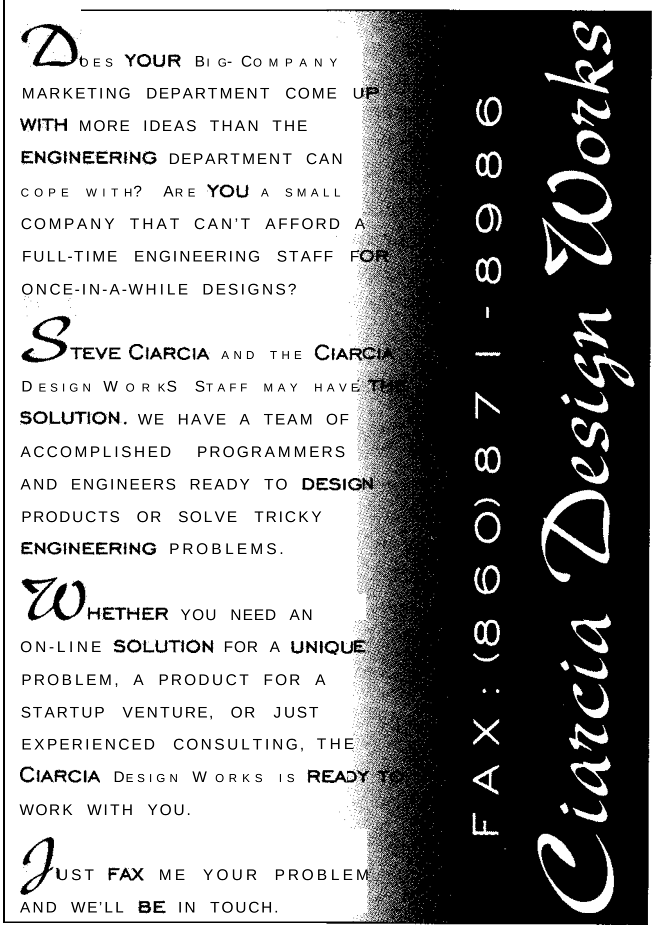
Also Available:

Professional Board (no programmer) $259
Hobbyist Pack (kit, no ZIF, '84 only) $199
Hobbyist Board (kit, no ZIF, '84 only, no programmer) $139
Teachers manual Call

Prices exclude shipping. Parallax is a trademark of Parallax, Inc PIC & Microchip are registered trademarks of Microchip Technology Inc.

Sirius microSystems
172 Harvard Road
Waterloo, ON N2J 3V3
CANADA
+1·519·886·4462 (ph.)
+1·519·886·4253 (fax)
support@siriusmicro.com
Visit our Web page: http://www.siriusmicro.com

#111



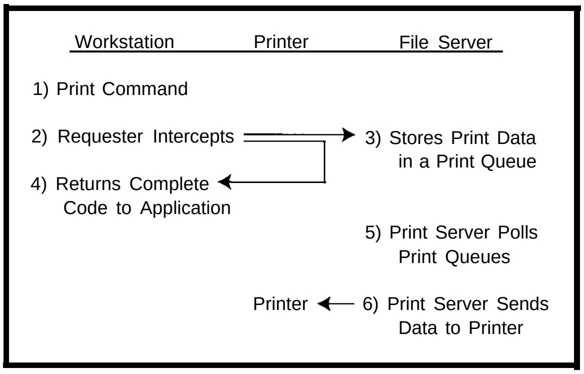


Figure 5—*This flow diagram details the actions involved in printing to a network-attached printer from a LAN workstation. Although the concept seems simple, three separate devices are involved. The file server is the coordinator for each device.*

BIOS system calls and send commands and data to the file server.

As Figure 5 illustrates, in a print operation, the application running on the workstation calls the BIOS system print routine. DOS Requester intercepts the call and redirects it to one of its software subroutines. This subroutine builds a print packet, which is sent to the file server.

The file server has a file defined as a print queue which the print data from the workstation is transferred into. When the workstation completes the print-data transfer to the file server's print queue, the file server returns a completion code to the workstation.

A background task on the file server periodically checks the print queues for data. When this so-called print server detects that data is present in the print queue, it transfers the queued data to a predefined hardware printer. You can check the status of this operation through a network file-server utility.

YOU NEED THE BASICS

In this article, I addressed the basics of network operation and provided a brief history of its evolution. I avoided the many esoteric concepts which make it seem like networks are based more on black magic than technology.

As time marches on, networking technologies will touch more and more aspects of our professional and personal lives. A basic understanding of the concepts behind this technology is essential to compete in the modern world. ❑

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

SOURCE

Novell
Novell, Inc.
1555 N. Technology Way
Orem, UT 84057-2399
(801) 222-6000
Fax: (801) 861-3933
http://www.novell.com/

IRS

401 Very Useful
402 Moderately Useful
403 Not Useful

#112

TCP/IP in Embedded Systems

FEATURE ARTICLE

Richard Ames

Transferring data between computers via TCP/IP extends beyond mainframes and desktops into the embedded world. Richard examines the layers of this protocol to see how data passes through the various systems.

I love the scene in sci-fi movies where the space hero introduces a trusty computer sidekick to a monolithic alien mainframe for a friendly file transfer.

Somehow, the hero manages to bring the two systems into close physical proximity, and through some magic and perhaps a bit of impromptu cabling, great volumes of information are transferred. When the data is later converted into a visually exciting graphical format, it turns out to be of critical value.

Now, I don't know how likely it is that we'll encounter alien beings piloting spaceships in another solar system. But, I do know that if I ever wanted to transfer information from a computer developed by an independent civilization, I'd start with TCP/IP.

TCP/IP is a suite of layered network-communication protocols best known as the communication protocols of the Internet. They are computer-architecture independent and implemented in an extensible, layered design. And, they have open standards with freely available specifications.

Traditionally, TCP/IP has been used on mainframes and UNIX workstations in universities and research centers for system-wide distribution of information. But in recent years, software using TCP/IP has been increasingly incorporated into PCs, giving desktop users access to the rich resources and capabilities they provide.

Today, it's also practical to incorporate these networking protocols in embedded systems with as little as 64 KB of address space. These systems can now become Internet citizens as well.

At its most basic level, the TCP/IP suite provides a well-designed method of transporting data between physically connected computers. Among the protocols, provisions for checksums and flow control keep error-free data flowing smoothly through the system.

And, since part of the motivation for developing it was to provide a nationwide computer network that could function during a nuclear attack, rest assured that reliability issues were well-considered.

In addition to the core protocols responsible for moving information, there are application-level protocols (e.g., ftp, telnet, and http), which are summarized in Table 1.

These protocols typically have a client component and a server component. A system running the client component can access the resources made available by systems running a server component.

For example, a Web browser is a client application that accesses the multimedia resources available on Web-server applications running on systems throughout the Internet.

Since the interaction between these client/server applications is specified in well-documented and -exercised protocols, the software you develop for your embedded system only needs to implement one end of the application. You can be sure that independently developed applications for the other end will work with your system.

For example, if you implement an ftp client in your embedded system, it should be able to download files from any ftp server in the connected network, even though you didn't develop the server. Standardized application protocols let you take advantage of off-the-shelf software and existing facilities in putting together networked client/server applications.

TCP/IP LAYERS

Let's take an in-depth look at the layers composing the TCP/IP suite. This set of layers is also known as a stack, since they can be described as being built one on top of the other.

Figure 1 shows two systems, A and B, each containing a TCP/IP stack and connected to a network. Figure 2 shows

Protocol	Application
smtp (Simple Mail Transfer Protocol)	Enables message transfer between systems
ftp (File Transfer Protocol)	Allows files to be moved between systems
telnet (Telecommunications Network Protocol)	Provides a terminal session with a system
http (HyperText Transfer Protocol)	Offers multimedia transfer between systems

Table 1—*Several different application protocols are used in a TCP/IP stack to provide common services.*

a number of protocols and the positions they occupy in the four-layer stack.

Certain relationships exist between the protocols. For example, ftp requires the transport layer to offer data-flow control. TCP provides this feature and UDP does not, so ftp applications must use TCP at the transport layer.

When an application running on system A needs to send information to a system-B application, it sends the information down through the stack. Each layer adds information to the data it receives from the layer above-a process known as encapsulation.

Figure 3 shows the terms used to describe data as it travels through the stack at each layer. The terminology depends on the protocol used at the stack's transport layer.

For example, an ftp application writes a stream of data and passes it to TCP at the transport layer. TCP adds its header to the data, creating a TCP segment, and passes the segment to IP at the network layer. IP adds an IP header, making an IP datagram, and passes the datagram to the link layer.

If the link layer is Ethernet, it adds an Ethernet header, creating a network frame. A device driver then readies the frame for transmission on the network.

To better understand the role each layer plays in the TCP/IP stack, consider the problem of moving file contents between systems A and B.

System A tells system B which file to transfer. Here, the task is covered by an application-level protocol. The ftp protocol, in which an ftp client application is running on system A, sends a RET R command to system B, which is running an ftp server application.

Following the ftp standard, RET R specifies the file for transfer using:

```
RETR filename <CR> <LF>
```

Before an ftp server accepts the RETR command, the client must establish a connection with the server and verify that it has permission to transfer files. But, let's assume that these steps have taken place.

Next, the ftp client application calls a write function, passing the text of the RET R command along with an identifier for the established connection.

The file-transfer processes need to be identified. Large systems supporting TCP/IP may run a multitasking OS that enables several tasks to communicate over the network simultaneously and independently.

To ensure that the information is delivered to the appropriate application, port numbers are associated with each end of an application. with ftp, the port number associated with the client is dynamically generated to be unique on the system when the session starts.

The server's port number is established as port 21 as part of the ftp standard. Port-number information is stored in the TCP header, which is inserted before the datastream written by the ftp application. This TCP segment then passes to the network layer.

The systems involved in the file transfer must also be identified, just as a link to the transfer processes is included in the transport-layer header. Thus, once the frame is ready to go out onto the network, it's clear where it should be delivered. And, once it is delivered, it's clear where it came from.

The network layer adds an IP header including the packet's sender and receiver addresses, which are four-byte Internet protocol addresses. This layer is also responsible for breaking up a datagram if it can't fit in one frame on the underlying link layer. So, it's possible that a large TCP segment is broken up into many fragments on its way down to the link layer.

Each system needs to be physically connected to a network. Of course, the computers don't need to be connected to the same physical wire. But, each computer must be connected to some form of network, and if the computers are on separate networks, there must be a path for data to travel between networks.

The network type and the format and interpretation of the frames appearing on the network are defined in the link layer. This layer provides a bridge between the network layer and the physical media carrying the information. The RET R command and the transport and network headers must be transmitted on the network in a way that makes this information distinct.

The link layer can be as simple as two RS-232C ports connected via a null modem cable. Other possible solutions include Ethernet, ARCnet, Token Ring, and FDDI.

The link layer must also handle physical addressing. In a point-to-point network, this is simple. All transmitted and received data travels directly between the local and remote interface.

But, information sent over a broadcast medium like Ethernet can be received by all interfaces connected to the network cable. So, these interfaces have hardware-supported mechanisms for recognizing frames that are directed to a particular interface.

If the Ethernet header contains the six-byte destination address matching the interface's physical address, then the interface receives the frame and alerts the processor.

Otherwise, the frame is allowed to zip by without disturbing the processor. A special broadcast address is also available. All interfaces receive frames directed to that address.

After the network frame is completely assembled, it is transmitted on the attached network. If system B is on the same network, the network frame

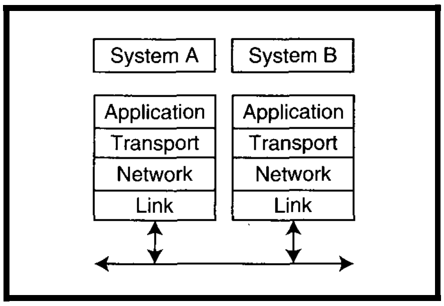


Figure 1—*The TCP/IP suite can be divided into* four layers.

arrives immediately. If system B is on another network, you need a router.

A router is a device containing more than one network interface. It joins networks by forwarding information between directly attached networks. If system A determines that system B is not on its own network, then system A sends the frame to router A. Router A is then responsible for delivering the information to its destination.

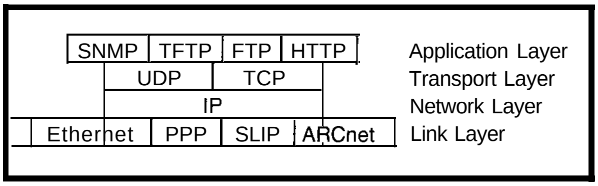


Figure 2—*The* protocol layers *are stacked on fop of each other.*

But, as shown in Figure 4, system B may not be on any of the networks attached to router A. In this case, the information goes to more routers until it arrives at system B's network.

When host B receives a network frame from system A, it first strips off the link-layer encapsulation, yielding a datagram. The destination address in the datagram is compared to system B's IP address to ensure that it was directed correctly.

If all is in order, the datagram encapsulation is stripped off to reveal a TCP segment. The transport layer receives this segment and checks the destination port number in the TCP header and finds that the segment is directed to port 21, the ftp port.

The remaining bytes intended for the application are then added to the datastream for the ftp server. The ftp server process examines the stream to see if system A sent a RETR command.

LINK-LAYER ISSUES

You have a number of options for linking an embedded system to a TCP/IP network.

The simplest is to use an asynchronous serial interface to establish a point-to-point link with another systern.

This option is attractive because existing system designs may already contain a spare serial port or one can be added without much effort. The physical media for such a link can be a three-conductor cable or a modem and telephone lines.

The link layer also includes a software component for bridging between the network interface and network layer. With a serial interface, the most common options are SLIP and PPP. Both ends of the point-to-point interface must run the same link-layer software to communicate.

SLIP (Serial Line Internet Protocol), an early addition to the TCP/IP suite, establishes a TCP/IP network link over a dial-up telephone line. It's fairly featureless compared to its successor, PPP (Point-to-Point Protocol). But, it's simple and gets the job done.

SLIP encapsulates IP datagrams with a flag character and then escapes the flag character when it occurs in the datagram with a special sequence. One difficulty with SLIP is its assumption that each end of the link is assigned an IP address before the link starts transferring data.

To address this issue, PPP includes a mechanism for negotiating the IP address of each end of the connection. PPP also includes a checksum and provides extensions such as compression and authentication.

These features make PPP more complicated in its implementation and configuration, but many users want at least some of these features. PPP is becoming a very popular standard.

Creating a network out of point-to-point links is possible, but if you connect a number of systems in a small area, it's more efficient to use a shared networking technology like Ethernet or ARCnet.

Ethernet and ARCnet controllers are fairly similar. Both systems provide shared networks that allow high-speed access for a number of systems. ARCnet tends to cost less and guarantees a maximum latency for network access. Ethernet is higher speed and more likely to be in place in an existing network.

ARCnet and Ethernet controllers need an interrupt to get the processor's attention when a frame is received from the network or when the transmit buffer is empty and frames are waiting to go out. These controllers are typically allocated a small space in the processor's I/O memory. Some controllers also support DMA transfer between the controller's buffers and system memory, as well as advanced buffer-management features.

A number of options exist for the physical media linking an Ethernet network. The two most common solutions are unshielded twisted-pair cable (10BaseT) and coaxial cable (10Base2).

Coaxial cable is also known as cheapernet because it can be a comparatively low-cost solution. Unshielded twisted-pair cable and coaxial cable are actually comparable in cost, but networks using 10BaseT must include an Ethernet hub as well as cabling.

10Base2 networks can connect directly from interface to interface without a hub. However, network traffic can be corrupted if a connection to an interface is disconnected. 10BaseT doesn't suffer from this difficulty.

BOOTING THE SYSTEM

The first thing a system running TCP/IP wants to know at boot time is its IP address. Without an IP address, the system can only send information through network broadcasts-not the most efficient way to network. If a system doesn't know its IP address when it boots, you can use the address 0.0.0.0, but only as part of the system initialization process.

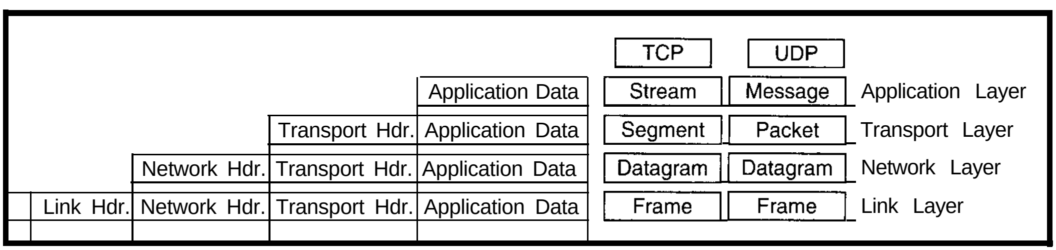


Figure 3—*Each layer in fhe stack has a special name for its information.*

Retrieving RFCs from the Internet

RFCs are documents that include the official standards for the protocols making up the TCP/IP suite. You also find proposals, recommendations, observations, and even poetry among them. RFCs (Requests For Comments) reflect how TCP/IP standards come into being. They are proposed and openly discussed, reviewed, and tested until they're recommended for advancement to the level of Internet standard. RFCs are also freely available via E-mail, ftp, or the Web.

For the latest information on obtaining RFCs, send E-mail with the subject getting rfcs to <rfc-info@isi.edu>. In the body of the message, type help: ways_to_get_rfcs. You'll receive a reply describing the process.

If you have ftp access, you can retrieve RFCs from the host <ds.internic.net> in the directory /rfc. Figure i shows an example anonymous ftp session from a UNIX system.

Note that rfc-index.txt is the first file retrieved—a good choice for the first file to download. All the RFCs are listed here in entries that specify the title, names of the authors, length, and other details. It's a good idea to look at the current RFC index first because the index entry indicates if a particular RFC has been updated or rendered obsolete. Then, the RFC can be retrieved using a filename of the form rfcxxxx.txt.

You can also use a Web browser to review the RFCs online or download them onto your hard disk. A good starting point is the Web page administered by the RFC editor at <http://www.isi.edu/rfc-editor/>.

Currently, more than 2000 RFCs are available, and copying them all to your hard disk could be rather time consuming. If you want all this information locally, consider purchasing a copy of the RFCs on a CD-ROM.

```
%ftp ds.internic.net
Connected to ds.internic.net.
220- InterNIC Directory and Database Services
220-
...
220 ds0 FTP server (V.4.105 Fri Jan 5 14:34:54 EST
  1996) ready.
Name (ds.internic.net:richard): anonymous
331 Guest login ok, send your email address'as
  password.
Password: <type in your email address here>
  230- Guest login ok access restrictions apply.
230- Local time is:'Mon Oct 28 20:19:43 1996
230
ftp> cd rfc
250 CWD command successful.
ftp> get rfc-index.txt
200 PORT command successful.
150 Opening ASCII mode data connection for rfc-i
  ndex.txt (274723 bytes).
226 Transfer complete.
local: rfc-index.txt remote: rfc-index.txt
281113 bytes received in 27 s (10 KB/s)
ftp> quit
221 Goodbye.
%
```

Figure i--The *RFC index can be retrieved using ftp.*

The IP address can be configured by hand and stored in nonvolatile memory, or the system can take advantage of several mechanisms that obtain this information from another system on the network (see Table 2). The latter approach allows IP address assignments to be centrally administered, which saves time and is less error prone.

BOOTP (BOOTstrap Protocol) not only provides an IP address but also delivers additional network-configuration information. It also sets the system up to receive a binary image across the network which can be used to augment the ROMed software the system is booted from.

There's a broader question concerning IP addresses, however. Will your system be part of a private network or the Internet?

For some applications, a stand-alone network is all you need. Such systems should adopt IP addresses from a block of addresses reserved specifically for private internets. One of these blocks is in the range 192.168.0.0-192.168.255.255.

You can use TCP/IP's many features and applications without worrying about where this network fits into the ever-crowded Internet. And, when you join the Internet, the IP addresses can be reconfigured to lie within Internet space.

If your embedded system joins the Internet, the question becomes: Where in IP space will you hang your systems! If the embedded systems are part of an established network, it may be practical to simply include support for any of the automatic mechanisms for retrieving an IP address on bootup and leave it to the system integrators to add your IP address to the central management system.

Protocol	Link Layer	Remarks
RARP	Ethernet	RARP server must be on the same network
IPCP	PPP	Good method for point-to-point links
BOOTP	any	Can deliver more information than just the IP address
DHCP	any	Allows IP address to expire after a lease time

Table 2—*These protocols can deliver an IP address af boot time.*

If the embedded system is for an organization that has no block of IP addresses to share, a local Internet Service Provider (ISP) will handle these needs for a monthly fee. Most dial-up PPP accounts are geared toward running client applications primarily because the ISP dynamically selects an IP address for the dial-up link when the connection is made.

An embedded system normally cannot run as a server on this type of link because the client end of the application doesn't know which IP address to contact to obtain services from the embedded system. ISPs often offer dedicated IP addresses at a higher rate, enabling a system to maintain a PPP link and run as a server.

The system's location has significance for both the physical media and the IP address assigned to a system. If the system operates in an office or factory, you might link the system with an existing network using whatever physical-layer media was adopted and part of the area's IP address

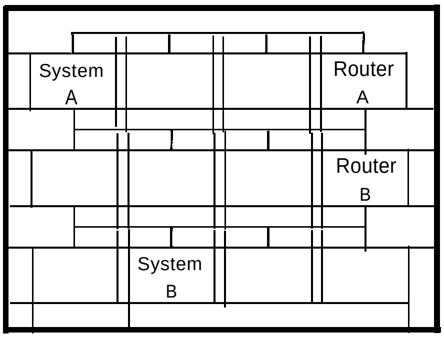


Figure 4—*System A can communicate with system B through router A and router B.*

space. Systems installed in an area without networking support can use telephone lines or radio links.

With the great benefit of worldwide access through the Internet comes the potential threat of someone using your system without permission. Security is usually an issue of a system running a server application.

Ftp and telnet servers typically start a session prompting for a user name and password. Although the user may not see the password on their terminal screen, it may be visible inside the packet delivered to the system running the server (there are ways to view packets as they stream across the Internet).

If your system needs an extra level of security, you can build it in via an encrypted key delivered across the Internet. The Internet Protocol V.6 provides facilities for security at the network level.

A TCP/IP stack is usually not something you can put together over a long weekend. Some publicly available implementations exist, notably the BSD Lite distribution. But, these tend to be rather extensive collections of software designed for the UNIX operating system and not easily adapted to most embedded systems.

If your system is an embedded PC, check out the vendors offering TCP/IP stacks as DLLs potentially suitable for your environment. There are also companies that offer a TCP/IP stack optimized for embedded environments.

REACHING OUT

In this overview of the TCP/IP protocol suite, I've presented the basics of some core protocols, as well as how they work together. In the future, I'm hoping to describe the implementation of a system using TCP/IP to enable an embedded system to interact with a Web browser.

If you're curious, all the Internet Standards are documented among the RFCs (see sidebar "Retrieving RFCs from the Internet"). And, there's a wealth of information in the networking section of your local technical bookstore.

Surely, there are more straightforward ways to transfer information between two computers than TCP/IP. The protocols' complexity is significant, and implementations of a TCP/IP network stack take a portion of a systems resources for memory space and processor cycles.

However, investing in support for TCP/IP protocols can extend the power and reach of an embedded system into a resource-filled network that's growing every day.

Richard Ames is a staff engineer at U.S. Software. He finds that working with networking software allows him to gather more computers around him than the average engineer. You may reach Richard at richard@ussw.com.

REFERENCES

J.B. Postel, "User Datagram Protocol," RFC 768, 1980.

J.B. Postel, ed., "Internet Protocol," RFC 791, 1981.

J.B. Postel, ed., "Transmission Control Protocol," RFC 793, 198 1.

J.B. Postel and J.K. Reynolds, "Telnet Protocol Specification," RFC 854, 1983.

W.R. Stevens, *TCP/IP Illustrated,* **1**, Addison-Wesley, Reading, MA, 1994.

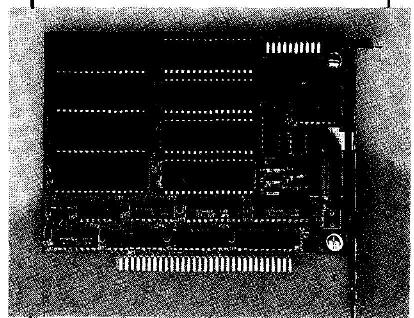
SOURCE

USNet
U.S. Software Corp.
14215 NW Science Park Dr.
Portland, OR 97229
(503) 641-8446
Fax: (503) 644-2413

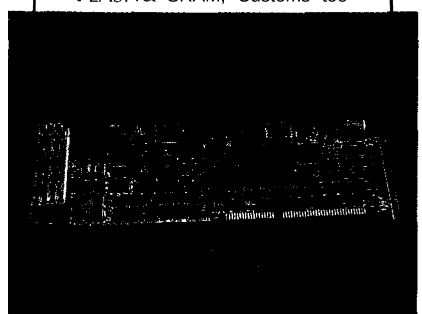
I R S

404 Very Useful
405 Moderately Useful
406 Not Useful

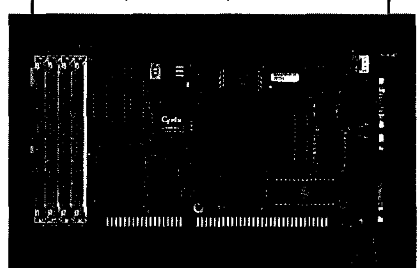
VMAX®
QUALITY PRODUCTS
RESPONSIBLE SERVICE
RELIABLE DELIVERY



ENHANCED SOLID STATE DRIVE —$105*
4M Total, Either Drive Bootable
½ Card 2 Disk Emulator
Flash System Software Included
FLASH & SRAM, Customs too



486 SLAVE PC — $898"
Add up to 4 Boards to One Host PC
Fast Data Transfer and I/O
PC-104 Port, IDE & Floppy Control
Independent Processors on One Bus
No Special Compilers Needed



486 66MHz SINGLE CARD COMPUTER — $335"
Up to 2.5Meg Flash/Sram drive
Compact-XT height ½ card size
Industry Standard PC-I 04 port
L2 cache to 64K—DRAM to 16Meg
Dual IDE/Floppy connectors

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

TEMPUSTECH, INC.
TEL: (800) 634-0701
FAX: (941) 643-4981
E-Mail: cpusales@tempustech.com
I-Net: www.tempustech.com

Fax for fast response!
295 Airport Road
Naples, FL 34104

#114

FEATURE ARTICLE

Jim Butler

BACnet

An Object-Oriented Network Protocol for Monitoring and Distributed Control

BACnet, unlike Ethernet or TCP/IP, focuses on the structure of the data communicated. Geared for industrial building systems, this standard offers functionality within a device via objects and services.

Local area networks have taken off in the desktop world. But, industrial controllers have very different needs than desktop machines.

In this article, I provide a technical overview of BACnet-an ANSI-standard network protocol for controlling and monitoring commercial and industrial building systems. BACnet also provides a potential solution for applications in a wide range of distributed-control and monitoring situations.

But first, let me give you some background. Automated systems in modern commercial or industrial buildings are much more complex than in homes, entailing more sophisticated controls and communication between various control devices.

Several different types of systems exist in these buildings-HVAC (heating, ventilation, and air conditioning) control, fire detection and alarm, fire suppression, access control (security), lighting control, and vertical transport (elevators and escalators).

Many systems employ some data communication between their elements, which can be very useful—especially during emergencies. But until recently, the communication protocols of these systems were always proprietary.

Because users had so much difficulty interconnecting microprocessor-based building-automation equipment from different manufacturers, the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) formed a committee to create a standard network protocol for building automation.

After many meetings and three public reviews of draft standards, BACnet was approved in 1995 as ASHRAE Standard 135-1995. Later that year, it was approved as an ANSI standard. It is also a European (CEN) "prestandard."

BACnet STANDARD

BACnet differs from popular network protocols (e.g., Ethernet and TCP/IP) in one very important way. It focuses on the structure of the data communicated between monitoring and control devices.

In contrast, Ethernet and TCP/IP transmit data between network devices. This difference marks them as noncompeting protocols with BACnet. In fact, Ethernet and TCP/IP can transmit BACnet messages between BACnet-compliant devices.

The BACnet Standard states that its purpose is "..to define communication services and protocols for computer equipment used for monitoring and control of HVAC&R and other building systems.. .thereby facilitating the application and use of digital control technology in buildings."

To reach this goal, the authors developed a standard representation for the data communicated between control and monitoring devices. The result was an object model, network services, and a powerful message-encoding scheme.

Analog Input, Analog Output, Analog Value
Binary Input, Binary Output, Binary Value
Calendar
Command
Device
Event Enrollment
File
Group
Loop'
Multistate Input, Multistate Output
Notification Class
Program
Schedule

Table 1—*BACnet defines a number of standard object types, and nonstandard types can be defined for individual applications.*

BACnet's creators didn't want it to block innovation, so the standard defines how building-automation manufacturers may add proprietary functionality in a way that doesn't damage BACnet's fundamental structure. Also, an ASHRAE committee regularly discusses proposed additions and changes to the standard.

But, what doesn't the BACnet Standard define? Although the standard's authors wanted to facilitate communication, they also wanted flexibility. So, BACnet does not define:

- what BACnet capability a device must have beyond a minimum level
- what functionality a device must make accessible to other devices
- APIs
- the representation of data within a device
- a device platform (i.e., operating system and specific hardware)

Although the standard is complex and lengthy, the authors hope rather inexpensive microprocessor-based control devices could be BACnet compliant.

The standard gives equipment designers a lot of flexibility in choosing how many features of the BACnet Standard their device will support. Already, BACnet-compliant controllers have been built using 8-bit microcontrollers [e.g., Motorola's 68HC05 and Intel's 805 1).

The standard doesn't define what functionality a BACnet-compliant device must have to be network accessible. Hence, you can create a device and protect the proprietary aspects of the design.

Suppose you come up with an innovative temperature-control algorithm. Your BACnet-compliant temperature-control device can allow other BACnet-compliant devices to set the desired temperature, while keeping the temperature-control algorithm completely hidden.

The BACnet Standard also does not define APIs, providing more freedom in constructing BACnet software libraries. This contrasts with other protocols such as TCP/IP (UNIX sockets and WinSock for Microsoft Windows), and Novell's Netware.

BACnet OBJECTS

Control and monitoring devices differ in function and implementation. But for devices that are accessible to other devices, it's necessary to have a common way to represent functionality. BACnet meets this need using objects.

A BACnet object is a data structure representing a functional element within a device. Most BACnet messages make direct or indirect reference to one or more BACnet objects.

A number of standard object types are defined in Table 1. Nonstandard object types can be defined for particular applications. A typical controller has several objects.

Every BACnet object consists of a collection of properties, each containing a value of a specified data type. Some properties may contain values not restricted to a single data type. All objects must contain at least three specific properties, and virtually any useful object contains other properties.

Every standard object type has a set of mandatory and optional properties. While optional properties may be omitted, nonstandard properties may be added.

All BACnet objects must have an:

- object identifier-a numeric value uniquely identifying an object within a device
- object name-a unique name for the object within a device
- object type-a numeric code identifying the object's type

To see the utility of objects, let's examine a popular standard object type, the Analog Input (see Table 2).

Mandatory Properties:	
Object Identifier	Analog Input, instance #2
Object Name	"Zone 2 Temperature"
Object Type	Analog Input
Present Value	70.0
Status Flags	In_Alarm = F, Fault = F, Overridden = F, Out_of_Service = F
Event State	NORMAL
Units	Degrees Fahrenheit
Optional Properties:	
Description	"temperature near the elevators on the second floor"
Device Type	"Thermistor"
Reliability	NO-FAULT-DETECTED
Out of Service	FALSE
Update Interval	10s
Min. Present Val.	0.0
Max. Present Val.	120.0
Resolution	0.5
COV Increment	2.0
Time Delay	15s
Notification Class	5
High Limit	84.0
Low Limit	So.0
Deadband	1.0
Limit Enable	HighLimit=T, LowLimit=T
Event Enable	ToOffnormal=T, ToFault=T, ToNormal=T
Ached Transitions	ToOffnormal=T, ToFault=T, ToNormal=T
Notify Type	EVENT

Table 2—*These* properties are contained in *an* Analog *Input* object.

An Analog Input object typically represents a sensor measuring some quantity (e.g., temperature or flow). The measured value is contained in the Present Value property of an Analog Input object. Naturally, this value changes over time. The other properties identify, describe, and supplement the Present Value.

In network protocols that aren't object oriented, the sensor is represented by a single value (e.g., an input register in the Modbus protocol), which is obviously simpler but much less useful. In BACnet I/O objects, several optional properties offer change-of-value or alarm notification.

A particularly important object type is the Device object. Any physical device with any BACnet objects must contain one Device object. The properties in a Device object contain general information about the physical device.

The Object List property is important because it lists all BACnet objects in that controller. The value of the Object Identifier property of a Device object must be unique throughout a BACnet network. 1'11 examine the implications of this later.

Interoperability between equipment from different companies is enhanced when engineers designing controllers make maximal use of standard and published nonstandard object types.

Device Profile object types can also enhance interoperability and interchangeability of components for a particular application and simplify network configuration. For example, an object type could be created to represent all of a thermostat's typical functionality.

The committee maintaining the BACnet Standard is actively discussing possible new standard object types for HVAC applications. New standard object types for other building-automation applications are likely to be added.

BACnet SERVICES

While BACnet objects abstractly represent network-accessible functionality in a device, its services enable application programs to access the functionality in other devices. When a service is invoked, a request message is transmitted. And, often, the request recipient transmits a response.

The BACnet Standard defines a broad set of services which cover most users' needs (see Table 3), but it also defines a way for devices to transmit proprietary messages.

The client-server model directs BACnet services. A client asks a server to perform some service. While servers are typically powerful computers in the office-automation world, BACnet servers may be as simple as a sensor.

Client and server roles are not fixed. A device may be a client in some transactions and a server in others.

Let's look at the WriteProperty service in detail. A client device invokes WriteProperty to change the value of a particular property in a specified object located in a server device.

From the client's standpoint, invoking WriteProperty is like calling a function with the arguments:

- destination address (mandatory)
- object identifier (mandatory)
- property identifier (mandatory)
- property array index (optional)
- property value (mandatory)
- priority level [optional)

To continue with the function-call analogy, WriteProperty returns a Result(+) if the client received a positive acknowledgment from the server or a Result() and error information if the WriteProperty request failed.

Listing 1—*The Modbus protocol uses a message format which is rigid but space efficient. Here's one example of a Modbus message format.*

```
query:
  slave address (8 bits)
  function code (06h)
  holding register address  16 bits
  preset data (16 bits)
  error check
response:
  slave address (8 bits)
  function code (06h)
  holding register address  16 bits
  preset data (16 bits)
  error check
```

When a device's application program invokes a real API function implementing WriteProperty, the device should transmit a WriteProperty request message to the server and wait for acknowledgment. Assuming the server receives the request, it acknowledges if the request was successfully acted on or sends an error message otherwise.

Invoking most services normally results in the transmission of both a request and response message. However, some services do not trigger responses (e.g., UnconfirmedEventNotification), and others are intended to be broadcasted (e.g., I Am, which announces the presence of a device on the network).

ENCODING MESSAGES

Traditional network protocols encode network messages using implicit encoding. The message parameters and parameter formats are fixed for a given message type.

The main advantages of implicit encoding are that it results in compact and easy-to-decode messages. However, these efficiencies are obtained at the expense of flexibility and may lead to the proliferation of message types over time as the protocol evolves.

For an example of implicit encoding, let's examine the Modbus protocol's Preset Single Register command shown in Listing 1. This encoding is compact. Such messages can be quickly decoded by the network protocol software.

But, consider the implications of this command. It can only be used to preset 16-bit holding registers, and it supports up to 65,536 holding registers in a device.

For the designers of Modbus, these assumptions must have seemed reasonable, and they may still be valid today. Protocol designers choose implicit encoding because compact encoding or fast-message decoding seems more important than flexibility.

Since BACnet needed to be both flexible and efficient, it combines implicit and explicit encoding.

Object-Access Services
- ReadProperty
- WriteProperty
- ReadPropertyMultiple
- WritePropertyMultiple
- ReadPropertyConditional
- AddListElement
- RemoveListElement
- CreateObject
- DeleteObject

Alarm and Event Services
- SubscribeCOV
- ConfirmedCOVNotification
- UnconfirmedCOVNotification
- ConfirmedEventNotification
- UnconfirmedEventNotification
- AcknowledgeAlarm
- GetAlarmSummary
- GetEnrollmentSummary

File-Access Services
- AtomicReadFile
- AtomicWriteFile

Remote Device Management Services
- DeviceCommunicationControl
- ConfirmedPrivateTransfer
- UnconfirmedPrivateTransfer
- ReinitializeDevice
- ConfirmedTextMessage
- UnconfirmedTextMessage
- TimeSynchronization
- Who-Has
- I-Have
- Who-Is
- I-Am

Virtual-Terminal Services
- VT-Open
- VT-Close
- VT-Data

Table 3—*BACnet defines a very flexible set of standard services that are invoked by a device needing to send or retrieve information from another device.*

Listing 2—*The format of a BACnet Confirmed Request message can be specified using ASN.1 notation.*

```
BACnet-Confirmed-Request-PDU ::= SEQUENCE {
  pdu-type                          [0] Unsigned (0..15), 0 for this PDU
  segmented-message                 [1] BOOLEAN,
  more-follows                      [2] BOOLEAN,
  segmented-response-accepted       [3] BOOLEAN,
  reserved                          [4] Unsigned (0..31), set to zero
  max-APDU-length-accepted          [5] Unsigned (0..15), encoded
  invokeID                          [6] Unsigned (0..255),
  sequence-number                   [7] Unsigned (0..255) OPT,
  proposed-window-size              [8] Unsigned (1..127) OPT,
  service-choice                    [9] BACnetConfirmedServiceChoice,
  service-request                   [10] BACnetConfirmedServiceRequest
                                         OPTIONAL
  }
```

BACnet's explicit encoding is based on the ASN.l Basic Encoding Rules (ISO standard 8825). The rules specify that extra bytes (called "tags") be embedded in a message to help the recipient decode the message.

A tag provides information about the message parameter immediately following it. A tag is a number that identifies which message parameter is present and the parameter's data type and length.

With tags, you can construct messages with optional parameters, which are quite common in BACnet. In addition, a single message type can easily accommodate parameters without a predetermined type or length.

To see what's possible with tags, consider BACnet's ReadPropertyMultiple service. It enables the client to request the values of multiple properties in multiple objects contained in a server.

The response message typically contains several property values having different data types and lengths. It's hard to imagine how to encode the response without tags or a similar mechanism.

Let's look at how BACnet messages are encoded. Listing 2 is the format of a Confirmed Request message written using ASN. 1 notation. It shows the message parameters in the order they appear with addressing information omitted.

Items [0] through [9] in the PDU (protocol data unit) constitute the message header, which is encoded implicitly (i.e., without tags). Item [10] (`service-request`) contains parameters specific to a particular service.

The `service-request` is encoded explicitly (i.e., with tags). In the case of a `WriteProperty` request, the `service-request` is as in Listing 3.

A `WriteProperty` request message has two optional parameters. Also, the data type of parameter `propertyValue` is unspecified (denoted by ABSTRACT-SYNTAX.&Type), allowing `WriteProperty` to accommodate properties of any data type.

The service-specific part of messages is explicitly encoded using tags. The values in square brackets ([0] through [4]), known as context tags, identify the message parameters.

The data type of parameter `propertyValue` is left undefined in the format specification of a `WriteProperty` request message. So, it's useful for a `WriteProperty` request to contain an additional tag (i.e., an application tag) that identifies the parameter's data type and length. It can then be interpreted by the recipient.

Listing 4 shows the bytes that could be present in a `WriteProperty` request, along with comments that identify and decode its elements.

As you can see, BACnet tags result in messages that are more complex to decode than messages exclusively using implicit encoding. However, tags enable BACnet to have a relatively small number of very flexible message types.

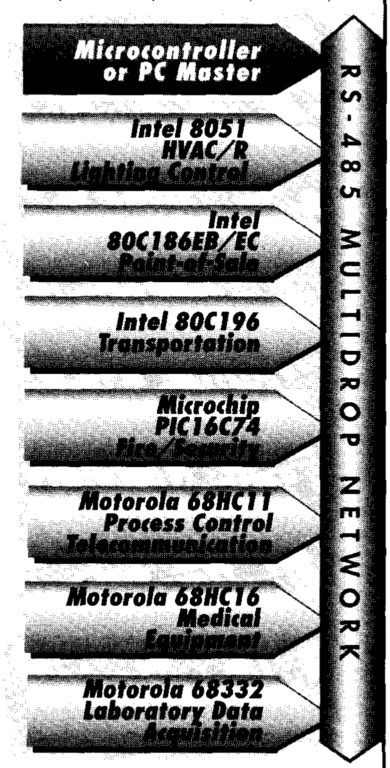
This design choice also enables new data types to be added without requiring the addition of new messages or changes to the format of existing messages.

Finally, Standard RS-485 Network Software

With Cimetrics' 9-Bit μLAN you can link together up to 250 of the most popular 8- and 16-bit microcontrollers (8051, 80C196, 80C186EB/EC, 68HC11, 68HC16, 68332, PIC16C74).

The 9-Bit μLAN is:

- ***Fast*** — A high speed (62.5k baud) multidrop master/ slave RS-485 network
- ***Flexible*** — Compatible with your microcontrollers
- ***Reliable***- Robust 16-bit CRC and sequence number error checking
- ***Efficient*** — Low microcontroller resource requirements (uses your chip's built-in serial port)
- ***Friendly***- Simple-to-use C and assembly language software libraries, with demonstration programs
- ***Complete*** — Includes network software, network monitor, and RS-485 hardware
- ***Standard*** — The 9-Bit μLAN is an asynchronous adaptation of IEEE 1118



Cimetrics TECHNOLOGY

e-mail: infoQcimetrics.com
55 Temple Place • Boston, MA 02111-1300 • Ph 617.350.7550 • Fx 617.350.7552

#115

MESSAGE TRANSMISSION

The BACnet Standard also specifies how BACnet messages are to be transmitted between devices connected to a BACnet network. Four data link/physical layer technologies have been officially sanctioned. Two are standard protocols:

- ISO 8802-3 (Ethernet) with ISO 8802-2
- ANSI 878.1 (ARCnet) with ISO 8802-2
- Echelon's LonTalk (using msg_send and msg_receive primitives)
- MS/TP (a token-passing RS-485 protocol developed specifically for BACnet)

In addition, BACnet developers anticipated the possibility of needing to connect multiple BACnet networks to form an internetwork. This required the creation of a network-layer protocol to allow the routing of BACnet messages between networks.

Routing is typically done via a special BACnet router device, but controllers connected to multiple BACnet networks may also perform this task.

In some cases, it's desirable to connect BACnet networks located a significant distance apart. The BACnet Standard defines two methods for handling this problem.

First, a special protocol permits the temporary or permanent connection of multiple BACnet networks using a simple serial point-to-point (PTP) communications link. PTP has also been frequently used to connect a PC workstation to a BACnet network over the telephone line.

The developers also defined methods by which a TCP/IP internetwork or Novell network could connect multiple BACnet networks using a technique known as tunneling.

The protocol's structure enables BACnet messages to be transmitted using many current or future networking technologies.

It could have sanctioned Token Ring as a data-link/physical layer protocol, but Token Ring was left out because it's not popular in the building-automation industry.

A current topic of discussion is how to enhance BACnet to make better use of TCP/IP internetworks.

Listing 3—The WriteProperty request is a type of BACnet Confirmed Service Request. It corresponds to what you see after the [10] in Listing 2.

```
WriteProperty-Request ::= SEQUENCE {
  objectIdentifier      [0] BACnetObjectIdentifier,
  propertyIdentifier    [1] BACnetPropertyIdentifier,
  propertyArrayIndex    [2] Unsigned OPTIONAL,
  propertyValue         [3] ABSTRACT-SYNTAX.&Type,
  priority              [4] Unsigned8 (1..16) OPTIONAL
```

ADDRESSING

Almost every communications network containing three or more connected devices must be able to identify devices. BACnet is no exception.

Every BACnet device may be uniquely identified on a given internetwork by the combination of its network number (two bytes) and data-link layer address (one to six bytes, depending on the network technology). BACnet routers have more than one such address.

BACnet requires that the object-identifier value of every Device object be unique throughout a BACnet internetwork, so it can be used as a kind of high-level address. From an implementation standpoint, the 32-bit Device object identifier is easier to use than the variable-length combination of the device's network number and MAC address.

NETWORK CONFIGURATION

Network configuration is a task which is very difficult to avoid entirely. However, BACnet has some features to help in the configuration process:

- taking an inventory of devices—BACnet provides the Who Is and I-Am services, which can be used to find out what devices are connected to a network, as well as the network address of each device found.
- locating a particular object-BACnet's Who-Has and I-Have services locate an object believed to be in a device connected to the network.
- determining device capabilities—Every BACnet device that contains any objects must have one Device object. This contains several properties describing the device's capabilities, including a list of all BACnet objects in it.

  The BACnet Standard also defines the format of a Protocol Implementation Conformance Statement, a document created by the device manufacturer to detail the capabilities of the device.
- configuring a device—BACnet's virtual terminal services allow a device to be configured over a network as if a terminal emulator was directly connected to the device. Virtual-terminal services might also be useful for performing diagnostics.

Listing 4—Here, I've constructed a real WriteProperty request. The hexadecimal bytes would actually be transmitted within a packet.

```
; msg header for a Confirmed Request (implicitly encoded)
00           ; Confirmed Request, unsegmented
04           ; max application msg size = 1024 bytes (from a table)
1D           ; invokeID = 29
0F           ; service-choice = WriteProperty Request

; msg body for WriteProperty Request (explicitly encoded with tags)
0C           ; context tag 0 (object identifier), length = 4 bytes
00 80 00 03 ; Analog Value object, instance.113
19           ; context tag 1 (property identifier), length = 1 byte
55           ; Present-Value property
3E           ; context tag 3 (property value), opening tag
44           ;  appl tag 4 (single prec real number), len = 4 bytes
43 34 00 00  ; 180.0 (value of present-value property, IEEE 754 enc)
3F           ; context tag 3 (property value), closing tag
```

BACnet's FUTURE

BACnet should evolve as technology advances. The standard has "continuous maintenance" status, so anyone can propose changes or additions at any time.

The BACnet committee meets semi-annually to debate proposed changes and to issue clarifications and errata. ASHRAE's rules ensure that approved substantive changes will not take effect until after a public review period has elapsed and all comments by interested persons have been carefully considered. Therefore, any changes happen rather slowly.

The BACnet committee is developing a conformance testing standard, creating device-specific object types for HVAC applications, and exploring ways to use BACnet on TCP/IP networks.

With increased interest from people working on applications other than HVAC, it's likely that the committee will consider additions to the standard in the near future.

BACnet seems likely to become the de facto standard protocol for automation systems in commercial and industrial buildings. Virtually every major HVAC-controls company has developed or is developing BACnet-compliant products, and already, it is used in hundreds of buildings in North America and Europe.

Some companies are adopting it as the native communication protocol for their devices, and others are developing gateways between BACnet and their proprietary communication protocols.

Manufacturers of fire-detection and access-control systems can use BACnet to increase the level of integration between different systems within a building or campus.

As well, BACnet has many attributes that match the needs of a much wider range of monitoring and distributed control applications (e.g., process control and factory automation).

I believe that BACnet's strongest features are:

- an object model that provides a standard network-visible representation of functionality present in control and monitoring devices
- a broad set of services
- a powerful message-codmg method
- the flexibility to use many different data-transmission technologies and media

BACnet isn't appropriate for all embedded applications which require networking. Simpler network protocols may be faster and have more predictable response time. They may also have a lower cost of implementation.

But, whatever your embedded networking application, BACnet deserves consideration. Unlike other standard protocols which have not yet entirely materialized, BACnet is field tested, clearly documented, and sufficiently flexible to satisfy many needs. ❑

Jim Butler is the director of software engineering at Cimetrics Technology. He's also a member of the committee in charge of maintaining the BACnet Standard. Jim received his B.S. and M.S. in engineering from M.I. T. You may reach him at info@cimetrics.com.

REFERENCES

R. Amborn and P. Ehrlich, "Decision Steps for Implementing a BACnet Interface Project," *ASHRAE Jrnl.*, **Nov. 1995,** 34–40.

ASHRAE, *ANSI/ASHRAE* **Standard** 135-1995, **1995.**

Modicon, *Modicon Modbus* **Protocol Reference Guide,** 1994.

H.M. Newman, **Direct Digital Control of Building Systems,** Wiley-Interscience, New York, 1994.

B. Swan, "The Language of BACnet," **Engineered Systems,** Jul. 1996, 24-32.

SOURCE

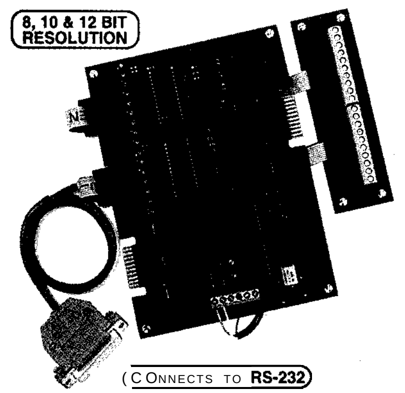
BACnet
Cimetrics Technology
55 Temple PI.
Boston, MA 0211 l-1300
(617) 350-7550
Fax: (617) 350-7552

I R S

407 Very Useful
408 Moderately Useful
409 Not Useful

AR-16 RELAY INTERFACE (16 channel) .... $ 69.95
Two 8 channel (TTL level) outputs are provided for connection to relay cards or other devices (expandable to 126 relays using EX-16 expansion cards). A variety of relays cards and relays are stocked. Call for more info.
AR-2 RELAY INTERFACE (2 relays, 10 amp)....$ 44.95
RP6 REED RELAY CARD (6 relays, 10 VA)......$ 49.95
RH-6 RELAY CARD (10 amp SPDT, 277 VAC)...$ 69.95

ANALOG TO DIGITAL



ADC-16 A/D CONVERTER* (16 channel/6 bit)..$ 99.95
ADC-8G A/D CONVERTER' (6 channel/IO bit).$124.90
Input voltage, amperage, pressure. energy usage, light, joysticks and a wide variety of other types of analog signals. RS-422/RS-485 available (lengths lo 4,000'). Call for info on other A/D configurations and 12 bii converters (terminal block and cable sold separately). Includes Data Acquisition software for Windows 95 or 3.1
ADC-8E TEMPERATURE INTERFACE' (6 ch)..$ 139.95
Includes term. block & 6 temp. sensors (-40° to 146' F).
STA-8 DIGITAL INTERFACE+ (6 channel).........$ 99.95
Input on/off status of relays, switches, HVAC equipment, security devices, keypads, and other devices.
PB-4 PORT SELECTOR (4 channels RS-422)....$ 79.95
Converts an RS-232 port into 4 selectable RS-422 ports.
CO-422 (RS-232 to RS-422 converter) .... $ 39.95

- EXPANDABLE...expand your interface to control and monitor up to 512 relays, up to 576 digital inputs, up to 126 analog inputs or up to 128 temperature inputs using the PS-4. EX-16, ST-32 &AD-l6 expansion cards.

*FULL TECHNICAL SUPPORT...provided over the telephone by our staff. Technical reference & disk including test software 8 programming examples in QuickBasic, GW Basic, Visual Basic, Visual C++, Turbo C, Assembly and others are provided.

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

. CONNECTS TO RS-232, RS-422 or RS-485...use with IBM and compatibles, Mac and most computers. All standard baud rates and protocols (50 lo 19.200 baud)

FREE INFORMATION PACKET...use our 600 number, Fax or E-mail to order, or visit our Internet on-line catalog.
URL: http://www.eeci.com
Technical Support (614) 464-4470

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

Internet E-mail: eeci1@ibm.net
International & Domestic FAX: (614) 464-9656
Use for information, technical support 8 orders.
ELECTRONIC ENERGY CONTROL, INC.
380 South Fifth Street. Suite 604
Columbus, Ohio 43215-5491

#116

Getting Pumped

FEATURE ARTICLE

Paul Paglia

DC-to-DC Charge-Pump Converters

If you decide to use a switch-mode power supply in a design, you need to know about DC-to-DC converter theory. It helps you make the best design choices and to eliminate unnecessary EMI.

Today's electronic equipment must be smaller and less expensive than ever before. The analog portions of many systems require special power supplies such as negative or higher voltages at relatively low currents (usually <100 mA).

But, while there are many ways to implement these power-supply solutions, only a few meet the requirements of low cost, high efficiency, and small size.

One way to generate these supply voltages is to combine dedicated voltage sources (e.g., negative rectification of the AC input, more batteries, special transformer windings, etc.) with linear regulators. However, extra batteries and transformer windings add cost and size. And, introducing a linear regulator reduces system efficiency.

In comparison, switch-mode power supplies offer more flexibility and higher efficiency. Switch-mode power supplies can step up, step down, or invert input voltages.

Figure 1 shows a common switch-mode power-supply scheme that steps up voltages. The step-up or "boost" switching regulator stores energy (supplied by V_{in}) in the inductor (L) when the FET is on. When it's off, the energy stored in the inductor (plus input energy] transfers to the capacitor.

The net result is an output voltage greater than the input voltage. The output voltage is fed back to the error amplifier through a preset resistor-divider network.

The error amplifier controls the on time of the FET through the PWM. This action indirectly regulates the output-voltage level by controlling how much energy is stored and transferred to the output.

In many applications, inductor-based topologies provide a good solution. However, they complicate matters by introducing radiated electromagnetic interference (EMI) as well as increased cost and size.

Implementing an inductor-based solution is not for every application, nor is it the most user-friendly solution. Selecting the proper inductor, rectifier, and storage capacitor requires some knowledge of DC-to-DC converter theory.

When you have large switching currents, you need to pay careful attention to board layout. When dedicated analog supplies are needed with low output current, DC-to-DC charge-pump voltage converters are the best solution.

These charge pumps provide up to 99% efficiency at a much lower cost than inductor-based solutions. Instead of using an inductor, two external capacitors store energy, reducing radiated EM1 and simplifying system design.

Offered in eight-pin surface-mount ICs and coupled with two capacitors (see Figure 2), the charge-pump converter provides a low-cost, easily implemented, small-outline solution. When light load, voltage inversion, or doubling is needed, charge-pump DC-to-DC converters are the best solution.

Let me introduce you to the concepts behind these devices. I'll discuss some of current trends and products.

PRINCIPLES OF OPERATION

The charge-pump DC-to-DC converter works on the principle of charge transfer. By storing charge in a "fly-

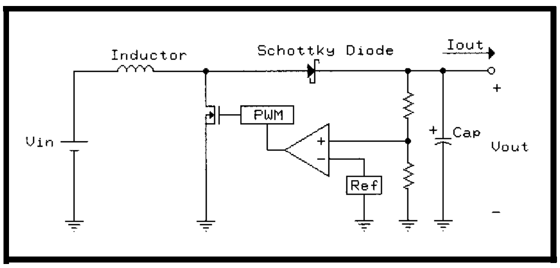


Figure 1—In this typical boost converter circuit, the converter's output is monitored via the resistor divider and fed back to the PWM circuitry. The PWM controls the on time of the FET.

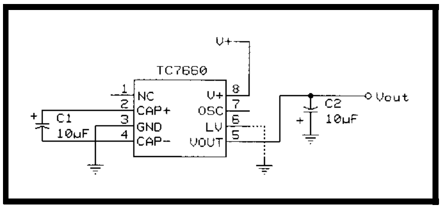


Figure 2—*TelCom's TC7660 is configured as a voltage inverter. Two external capacitors generate a negative voltage source. This voltage inverter configuration is the most common application for the charge-pump converter.*

ing capacitor" and transferring this charge to a reservoir capacitor, the DC-to-DC converter can perform common applications like voltage inversion and voltage doubling.

Figure 3 shows an example of the voltage inverter. An internal oscillator and logic circuit controls a switch matrix consisting of S1 and S2.

During the first half cycle, both switches are to the left. The flying capacitor (Cp) charges to the input voltage, V_{in}. During the second half cycle, the switches move to the right.

This shift ties the positive side of Cp to ground and the negative side of CP to V_{out}, offsetting Cp by $-V_{in}$. Cp and the reservoir capacitor (Cr) are now tied together in parallel, and charge is transferred from Cp to Cr.

For now, assume that switch losses are negligible. Under these conditions, the converter operates at close to 99% efficiency. The only losses are due to the supply current running the oscillator and logic circuitry.

With these assumptions, the converter can be modeled as an ideal voltage source. But, for a more realistic model, you must consider other things.

REAL-WORLD ANALYSIS

There is one P-channel (SW1) and three N-channel (SW2, SW3, and SW4) MOS power switches that allow charge to be transferred from the flying capacitor to the reservoir capacitor as shown in Figure 4. These switches are driven by the outputs of the logic circuitry, which are synchronized to half the internal oscillator frequency.

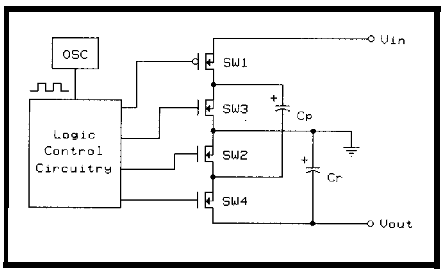


Figure 4-*This circuit shows the connections of the internal charge-pump switches (SW1–SW4) to the external capacitors (Cp and Cr) in a voltage inverter configuration. When SW1 and SW2 are on, Cp is charged to V_{in}. When SW3 and SW4 are on, Cp transfers charge to Cr and the load.*

CHARGING THE FLYING CAP

During the first half cycle, SW1 and SW2 are on and SW3 and SW4 are off. The circuit can be simplified to the one in Figure 5. The flying capacitor is charged to a voltage (V_{Cp}) based on equation (1). The total charge on the flying cap (q_p) is equal to CpV_{Cp}:

$$V_{Cp} = V_{in} - V_{in}\, e^{-(1/RCp)} + V_{Cp0}\, e^{-(1/RCp)} \quad (1)$$

Given that V_{Cp0} equals V_{Cp} at the beginning of the cycle, V_{in} is the supply voltage,

$$R = R_{SW1} + R_{SW2} + ESR_{Cp}$$

and

$$t = 0.5\left(\frac{1}{f_{charge}}\right) = \frac{1}{f_{osc}}$$

where f_{charge} equals 0.5 x f_{osc}.

SUPPLYING ENERGY TO LOAD

In addition to Cp being charged during this half cycle, Cr supplies current to the load as shown in Figure 6. At the beginning of this half cycle, Cr switches from being charged to supplying current to the load.

This represents a change in current equal in magnitude to $2I_L$. The voltage on the output of the converter (pin 5) drops instantaneously at the beginning of this half cycle due to the IR drop across Cr. It is reflected by $2I_L ESRCr$ in equation (2).

Assuming a constant current load condition, the discharge of Cr can be represented by equation (2). This equation can be used to approximate the devices output-ripple voltage:

$$V_{Cr} = V_{Cr0} - 2I_L ESRCr - \frac{I_L}{Cr}\,dt \quad (2)$$

where V_{Cr0} is the voltage on Cr at time 0, I_L is the load current, and

$$dt = \frac{1}{f_{osc}}$$

ESRCr directly contributes to the device's output-ripple voltage. If the device supplies a load of 20 mA and the capacitor has an ESR of 10 Ω, expect a minimum output ripple of 200 mV.

Increasing f_{osc} decreases the output-ripple voltage. However, increasing f_{osc} results in increased switching loss (increased I_{supply}). At some point, switching losses become significant enough that a dropoff in efficiency negates the benefits of increasing the operating frequency.

Increasing f_{osc} also enables you to use smaller capacitors. Keep in mind that reduction in capacitor size typically results in increased ESR. Such a reduction can detrimentally affect output ripple.

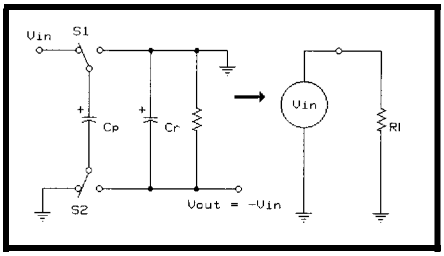


Figure 3—*This simplified circuit shows how the charge pump generates a negative voltage supply. V_{in} is stored on Cp, inverted, and transferred to Cr. Notice that the anode of Cr is grounded. When switching losses and capacitor ESR are ignored as in this circuit, the charge pump can be modeled as an ideal voltage source.*

TRANSFERRING CHARGE

During the second phase of the cycle, SW1 and SW2 are turned off and SW3 and SW4 are turned on. The simplified circuit is shown in Figure 7. The logic timing is such that a finite delay ensures that there's never a continuous path from V_{in} to ground.

Cp is tied in parallel with Cr and transfers a portion of its charge into Cr, as well as supplying current to the load. The voltage on Cp changes from V1 prior to the switch to V2 at the end of the cycle (see Figure 8).

The voltage V2 depends on the energy dissipated through circuit switch resistance and capacitor ESR, as well as the energy delivered to the load. The energy lost is defined as:

$$E = 0.5\ Cr\ (V1^2 - V2^2)$$

The amount of charge that was transferred, $q_{transfer}$, from Cp to the output is:

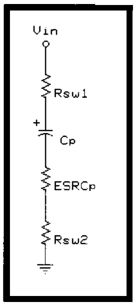


Figure 5—While *Cp* is being charged, the *ESR* of capacitor *Cp* and the resistance of SW1 and SW2 cannot be ignored. These components directly affect device power-conversion efficiency. Using low-ESR capacitors ensures high-efficiency operation.

$$q_{transfer} = CpV1 - CpV2 = Cp\Delta V_{Cp}$$

Knowing that the switch is operating at f_{osc}, an expression for current can be written as:

$$I = C\frac{dV}{dt} = Cp\frac{V1 - V2}{dt}$$

where

$$dt = \frac{1}{f_{osc}}$$

$$I = Cp\frac{V1 - V2}{\left(\frac{1}{f_{osc}}\right)}$$

or

$$I = \frac{V1 - V2}{\left(\frac{1}{f_{osc}}\right)CP}$$

Therefore, you can express the equivalent impedance during the transfer cycle as:

$$R_{transfer} = \left(\frac{1}{f_{osc}}\right)CP$$

OUTPUT IMPEDANCE

Taking into account the switch resistance and capacitor ESR, I can generate a more accurate model for the charge pump as an ideal voltage source in series with a resistor (see Figure 9).

This resistor, R,, represents the devices output resistance and is composed of:

- series resistance of R_{SW1}, R_{SW2}, and ESRCl when Cp is charging
- series resistance of R_{SW3}, R_{SW4}, and ESRCl when Cp is transferring charge to Cr
- the equivalent resistance during the charge transfer cycle
- the equivalent series resistance of the reservoir capacitor, ESRCr

This expression is summarized as:

$$R_0 = 2(ESRCp + R_{SW1} + R_{SW2}) + 2(ESRCp + R_{SW3} + R_{SW4}) + R_{transfer} + ESRCr$$

where

$$R_{transfer} = \left(\frac{1}{f_{osc}}\right)Cp$$

Note that because the switches are only on for half a cycle, the average resistance is multiplied by two. Assuming that the switch resistances are equal, R_0 simplifies to:

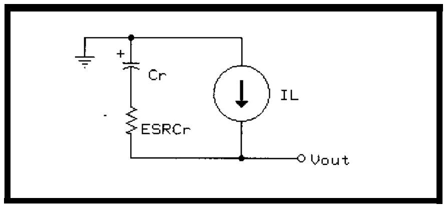


Figure 6—When *Cr* is supplying current to the load, the capacitor's *ESR* directly affects output-voltage level. Because Cr is tied directly to the load during this phase, switching losses aren't a factor.

ENGINEERING PROFESSORS LISTEN UP!

The reason you read *Circuit Cellar INK* is obvious.
Like thousands of others, you have come to realize that *Circuit Cellar INK* presents technically relevant computer-based applications in a comprehensive and instructive manner. In fact, our projects are presented just the way you'd do them if only you had the time.

Since 1994, we have provided thousands of college students with industry-quality development tools and information. This program is **FREE** and because of the continued success we have experienced, we'd like to extend our offer to the 1996-1997 school year.

If you're an engineering professor or teach qualified technical students, we'd like to give you and all of your students free subscriptions to *Circuit Cellar INK*. Please tell us how we can contact you directly so you don't miss a single issue. Just a reminder:
IF YOU WERE A PARTICIPATING COLLEGE LAST YEAR, PLEASE BE SURE TO UPDATE US WITH YOUR CURRENT ROSTER EACH SEMESTER.

WE'RE HELPING AMERICA REGAIN THE COMPETITIVE EDGE. WE'RE HOPING YOU WILL TOO!

Rose Mansella, Circulation Coordinator
Circuit Cellar INK
4 Park St.
Vernon, CT 06066

Tel: (860) 875-2199 Fax: (860) 871-0411
Internet: rose.mansella@circellar.com
World Wide Web: http://www.circellar.com/

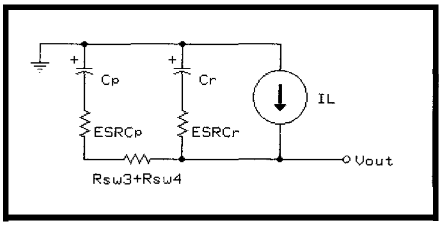


Figure 7—*This circuit identifies switch (SW3 and SW4) resistance as well as the capacitor ESR during the charge-transfer phase. During this phase, charge is* transferred from Cp to Cr and *the* load.

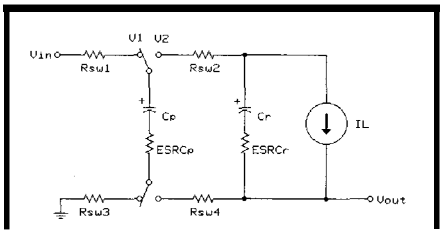


Figure 8—*This circuit identifies ESR and switch-resistance components associated with the charge-pump* converter. The lower the internal switch *resistance, the higher the output load capability*

$$R_0 = 4R_{SW} + 4ESRCp + R_{transfer} + ESRCr$$

where

$$R_{transfer} = \left(\frac{1}{f_{osc}} \right) CP$$

The switch resistance is a function of the transistor's geometry and process technology. You have only little control over its value, but you should be aware that R_{SW} is sensitive to temperature and supply voltage.

R_{SW} decreases with increasing supply voltage or decreasing temperature. Because R_{SW} is multiplied by four, increasing R_{SW} can have a significant effect on overall output resistance.

Looking at the remaining expressions that make up R,, we know:

- ESRCp has a four times larger effect on R_0 than ESRCr
- lowering f_{osc} increases R_0
- increasing f_{osc} decreases R_0 to a point. Eventually, ESR dominates, nullifying further increases in f_{osc}.

CAPACITOR SELECTION

The two major factors contributing to the charge pumps' output resistance are the IC's internal switch resistance and the ESR of capacitors Cl and C2. The internal switch resistance must be accepted as is. However, you must minimize the ESR of Cl and C2 to ensure optimal device operation.

You should select the external capacitor value and type that best suit your size and cost requirements. Ceramic capacitors offer the lowest ESR, followed by (in order of increasing ESR) OS-CON, film, Aluminum Electrolytics, and Tantalum capacitors.

Within one technology, ESR tends to closely track physical capacitor volume, given constant capacitance value and voltage rating. Consequently, lower ESR is usually obtained at the cost of increased capacitor size.

Film capacitors are obviously not a viable solution due to their excessive cost and size. Ceramic capacitors offer the lowest ESR, but they too can be cost prohibitive.

Their relatively low volumetric efficiency can result in size restrictions. Also, they exhibit an order of magnitude increase in ESR as operating temperature drops from +25°C to -55°C.

OS-CON capacitors offer an ESR only slightly higher than ceramics, but they consume more volume. The OS-CON capacitor has excellent linearity from -55°C to +125°C. Aluminum Electrolytics are ideal for low-cost applications where board space is not an issue.

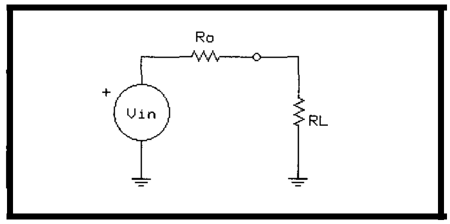


Figure 9—*The real-world model of the charge-pump voltage converter is depicted as a voltage source in series with output* impedance. *The output impedance is* a combination of internal switch resistance and external *capacitor ESR.*

Electrolytics like OS-CON capacitors are typically offered in a radial lead package but are available in surface-mount styles. Tantalum capacitors offer an ESR similar to the Aluminum Electrolytic, but their excellent volumetric efficiency provides the smallest profile solution at a reasonable cost.

Table 1 identifies some of the suppliers of low-ESR capacitors. Most manufacturers do not identify ESR. Instead, they provide DF (dissipation factor) specifications at 120 Hz. The lower the DF, the lower the ESR, since ESR is derived from:

$$ESR = \frac{DF}{2\pi fC}$$

LAYOUT

To ensure good noise protection, use these basic layout practices:

- minimize stray inductance by keeping board trace lengths to a minimum
- minimize ground impedance by employing a ground plane
- mount the charge-pump IC as close to the load as possible to minimize output impedance
- mount the supply decoupling and pump capacitors as close as possible to the charge-pump IC

Manufacturer	Series	DF @ 120 Hz	ESR @ 100 kHz	Cap (µF)	Working Voltage	Size	Capacitor Type
Chemi-Con	MV	0.12	–		50		
Sanyo	C'CBiS	0.14	–	4.7 4.7	25	4.3 3.0 x 4.3 x 5.2 x 5.4	SMT SMT Al Al Elec Elec
Panasonic	SM	0.06 0.07	0.150 Ω	4.7 4.7	16	7.9 7.0 x x 5.3 7.0 x x 3.3 9.0	SMT SMT Spec OS-CON Polymer
Sanyo					20		
Marcon	THC	<0.05	0.05 Ω	4.7	25	3.2 x 2.5 x 2.0	Ceramic
Marcon	TCC	co.05	–	4.7	25	5.6 x 5.0 x 2.5	Ceramic
Mallory			–				Ta
Matsuo	T2497E5	0.06	0.8 Ω	4.7	20	8.0 x 3.8 x 4.9	Ta
Illinois Cap	FHFA	0.12	4 Ω	4.7	50	5.0x11.0	Rad Ld Al Elec
Panasonic	SG	0.05 0.08	0.125 Ω	4.7 4.7	25	6.3 6.3 x x 11.2 6.8	Rad Rad Ld Ld OS-CON Al Elec
Sanyo							
Sanyo	SL	0.06	0.150 Ω	4.7		6.3 x 5.0	Rad Ld OS-CON
Panasonic	MF				25		
Matsuo	202	0.06 0.06	--	4.7 4.7	25 25	3.7 4.8 x 4.1 x 7.5 x 5.6	Rad Rad Ld Ld Tant Tant

Table I--Many low *ESR capacitor opfions are available for both surface-mount and fhrough-hole applications. The designer's task is to select the lowest ESR capacitor fhaf meefs the system's cost and size restrictions.*

PRODUCT OVERVIEW

TelCom offers 11 charge-pump DC-to-DC converters (see Table 2). These products

	TC660	TC962	TC1044S	TC7660	TC7660S	TC7660H	TC7662A	TC7662B	TCM680	TC682	TCM850-3
Pi" #											
1	Boost	Zener	Boost	N/C	Boost	N/C	N/C	Boost	C1–	C1–	C l+
2	C+	C+	C+	C+	C+	C+	C+	C+	C2+	C2+	C -
3	GND	GND	GND	GND	GND	GND	GND	GND	C2–	C2–	NegOut
4	C -	C -	C -	C -	C -	C–	C -	C -	V_{out}–	N/C	SHDN/OSC
5	V_{out}	V_{out}	V_{out}	V_{out}	V_{out}	V_{out}	V_{out}	V_{out}	GND	GND	FB/CONT
6	LV	FreqX2	LV	LV	LV	LV	N/C	LV	V_{in}	V_{in}	OUT
7	o s c	o s c	o s c	o s c	o s c	o s c	o s c	o s c	C l+	C l+	GND
8	v +	v +	v +	v +	v +	v +	v +	v +	V_{out}+	V_{out}+	IN
Primary Application											
8PDIP C,E	X	X	X	X	X	X	X	X	X	X	
8S0 C,E	X		X	X	X	X	X	X	X	X	X
8CDIP I,M		X	X	X	X				X	X	
16SO C,E		X									
Features											
V_{in} (V)	1.5–5	3-18	1.5-12	1.5-10	1.5–12	1.5–10	3–18	1.5-I 5	z-5.5	2.0-5.5	5-I 0
f_{osc} (kHz)	10	12	10	10	10	120	12	10	20	12	100
f_{osc} boost (kHz)	90	24	45	N/A	45	N/A	N/A	35	N/Z	N/A	N/A
OSC over drive	X	X	X	X	X	X	X	X			
Zener ref.		X									
LV Pin	X		X	X	X	X		X			
R_0 (Ω)	6.5	32	60	70	60	55	40	65	140	140	N/A
I_{supply} (max, µA)	200	190	80	80	80	460	190	80	500	185	2000
I_{shdn} (µA)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	I - 3
Summary	Highest current (100 mA), Limited V_{in}	High current (80 mA), 6.2-V reference, Wide V_{in} range	Medium current (20 mA), Extended 7660 V_{in}, Boost pin	Low current (20 mA), Low V_{in}, No boost	Medium current (20 mA), Extended 7600 V_{in}, Boost pin	High frequency (20 mA), Small-value capacitors	Medium current (40 mA), TC962 w/o ref or FreqX2	Medium current (20 mA), 1044S 7660S w/ extended V_{in} and slightly higher Ron	LOW current (10 mA), Positive and inverting voltage doubler	LOW current (10 mA), Inverting voltage doubler	LOW current (<5 mA), Regulated voltage inverter w/ shutdown

Table 2-*Given the* wide range of charge-pump devices available, you'll *likely* find one that fits your application. *Note the* variations in *supply current*, oscillator frequency, and output drive capability

are ideal for low-current, voltage-inversion, or doubling applications.

All of TelCom's charge-pump devices (except the TCM85*x*, TCM680, and TC682) are spin-offs of the TC7660. Except for the variations in output load capability, supply voltage range, supply current, and so on, they operate using the same principles.

The TCM680 and TC682 are unique because they provide positive and negative voltage doubling functions. In other words, the TCM680 can generate ±10 V from a single 5-V input.

The TCM85*x* family of charge pumps integrates a low dropout regulator, which provides a regulated negative output. The TCM85*x* is designed for applications where a low-noise regulated negative output is needed (e.g., gate bias for GaAs transistors in cellular phones).

PRODUCT TRENDS

Charge pumps were once considered unregulated DC-to-DC converters and used strictly for doubling or inverting voltages. And when it came to adjusting and regulating output, they fell short of inductor-based switchers.

Meshing the benefits of the charge pump with those of inductor-based converters, however, provides the ideal solution. A high-efficiency, adjustable, regulated converter with no radiated EM1 is achieved by integrating a linear regulator or other feedback circuitry into the charge pump.

Many charge pumps already have shut-down capability. The appearance of other features that further reduce the average device's power consumption is inevitable. Charge-pump oscillator frequencies are being decreased or increased to reduce supply currents or accommodate small capacitors.

As with most products, manufacturers are offering the same or more functionality in a smaller package. Consequently, package size is reducing from the standard eight-pin SOIC to MSOP or SOT-23-5 packages.

Smaller packages typically mean reducing die size, resulting in smaller output transistors and larger output resistance. Obviously, maintaining the same efficiency when going to a small package is the real challenge.

BEST FIT?

The charge pump is by no means all things to all people, but for certain power-conversion applications, it's the ideal solution.

The charge pump can't provide the output current and regulation of inductor-based DC-to-DC converters. But, it does provide a noise-free (EMI) power-conversion solution that's easy to implement, low cost, and compact.

These features combined with excellent efficiency at light loads, make the charge pump *the* converter of choice for many low-power applications. ❑

Paul Paglia is a field applications engineer for TelCom Semiconductor. He is a graduate of Tufts University and has worked in the semiconductor industry for 10 years. You may reach Paul at pags@mindspring.com.

SOURCE

DC-to-DC charge-pump converters, TC7660EV and TCM-850EV evaluation kits
TelCom Semiconductor, Inc.
1300 Terra Bella Ave.
Mountain View, CA 94043- 1800
(415) 968-9241
Fax: (415) 967-1590

I R S

410 Very Useful
411 Moderately Useful
412 Not Useful

MICRO SERIES

Joe DiBartolomeo

Standards for Electromagnetic Compliance Testing

Introducing Concepts and Standards

Part 1 of 4

According to Joe, the better designers understand FCC and European regulations of EMI and EMC, the more time and money they save. Find out what you've been missing.

In the past 20 years, the use of the electromagnetic spectrum has changed dramatically. Far more cell phones and computers now operate at much higher frequencies and power levels.

The average portable computer has a master clock above 100 MHz, causing harmonics into the gigahertz. The latest cell phones operate at 900 MHz.

Designers of electronic equipment must consider electromagnetic interference (EMI) and electromagnetic compatibility (EMC) issues like never before. However, the mere mention of EM1 and EMC to engineers usually brings moans, groans, colorful language, and a horror story or two.

Few areas in the field of electronics evoke such emotions. And, it's unfortunate since EMI/EMC issues and their associated tests have become de facto design standards for all electronic equipment.

A major reason for such negative reactions is the lack of understanding by electronics designers of the EMI/ EMC rules, regulations, standards, and tests. (From now on, I'll refer to rules, regulations, and standards simply as standards.)

This lack of understanding is particularly true of digital designers. When CPU clocks rates were in the low megahertz, there was a clear distinction between analog and digital. Digital designers had little cause to be concerned with EMI/EMC. But with ever-increasing clock rates, they too must pay attention to EMI/EMC.

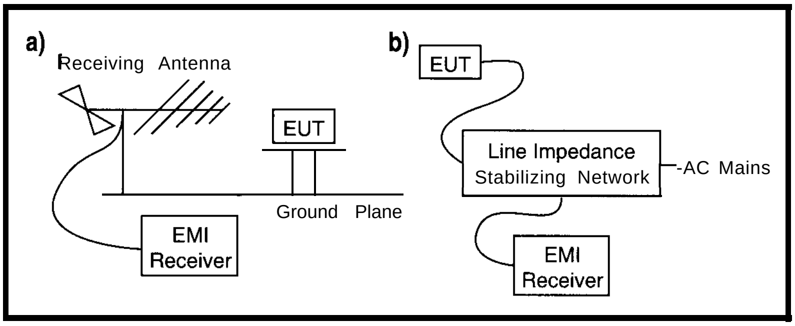


Figure 1--The *basic test setup for radiated (a) and conducted (b) emissions tests are fairly simple. These tests will be discussed further in* INK *80.*

The number of standards and tests associated with EM1 and compliance is numbing. It's difficult to determine which standards and tests apply to a particular piece of equipment.

In this series, I discuss only the standards and tests for equipment where a computer, microprocessor, and/or digital circuits (I refer to all of these as digital circuits] are a major source of EM emissions.

I ignore the standards and tests for such equipment as broadcast and telecommunications, although they most certainly contain digital circuits. Also, I only discuss the standards and tests demanded by the FCC and European agencies since they cover most of the industrial world.

And, this series is written for the design and development engineer. I'm not a compliance engineer, so I'll omit the nitty-gritty bureaucratic details, as well as the esoteric exceptions.

While I emphasize the technical issues of the standards and tests, don't use me as your sole guide to EMI/EMC compliance. This is intended a general resource. It in no way replaces the actual standards and tests.

Depending on your specific equipment and application, it may be worth it to read the actual standards and tests. However, let me give you some tips.

Do not read the standards at work. You will fall asleep and get fired. Read them at home, but not in bed (unless you suffer from insomnia).

The best way to read the standards and tests is standing up. Ambient temperature is also important. Turn down the heat. Or, if live in an area that requires air conditioning in winter, turn down the AC and start reading.

FCC

The FCC regulations that deal with equipment containing digital circuits, where they're the major source of EM emissions, are the *Code of Federal Regulations* Title 47 (CFR 47) Parts 15 and 18.

Part 15, Radio Frequency Devices, is the part that most equipment containing digital circuits normally seeks compliance under. Part 18, Industrial, Scientific, and Medical devices (ISM), covers a special class of products, most containing digital circuits.

Part 15 is divided into four subparts: general [A), unintentional radiators (B), intentional radiators (C), and unlicensed personal communications service (UPCS) devices (D).

Subpart A covers general topics and gives definitions. Subpart D deals with personal communications devices that operate in the 1910–1930- and 2390-2400-MHz frequency bands, which I won't discuss here.

Subpart C handles intentional radiators, which are usually broadcast and telecommunications equipment. The vast majority of products that contain digital circuits, where the digital circuits are the main source of EM emissions, are typically unintentional radiators and are covered in Subpart B.

Under Part 15, a piece of equipment is either an intentional or unintentional radiator. An intentional radiator is a "device that intentionally generates and emits radio frequency energy by radiation or induction."

These devices radiate RF energy as an inherent part of their functionality [e.g., cell phones, CB radios, and broadcast transmitters). While most of these devices contain digital circuits, they're not normally the main source of EM emissions. So, I won't discuss intentional radiators further.

An unintentional radiator is "a device that intentionally generates radio frequency energy for use within the device, or that sends radio frequency signals by conduction to associated equipment via connecting wiring, but which is not intended to emit RF energy by radiation or induction."

Virtually every electronic device produces unintentional emissions. Any digital circuit

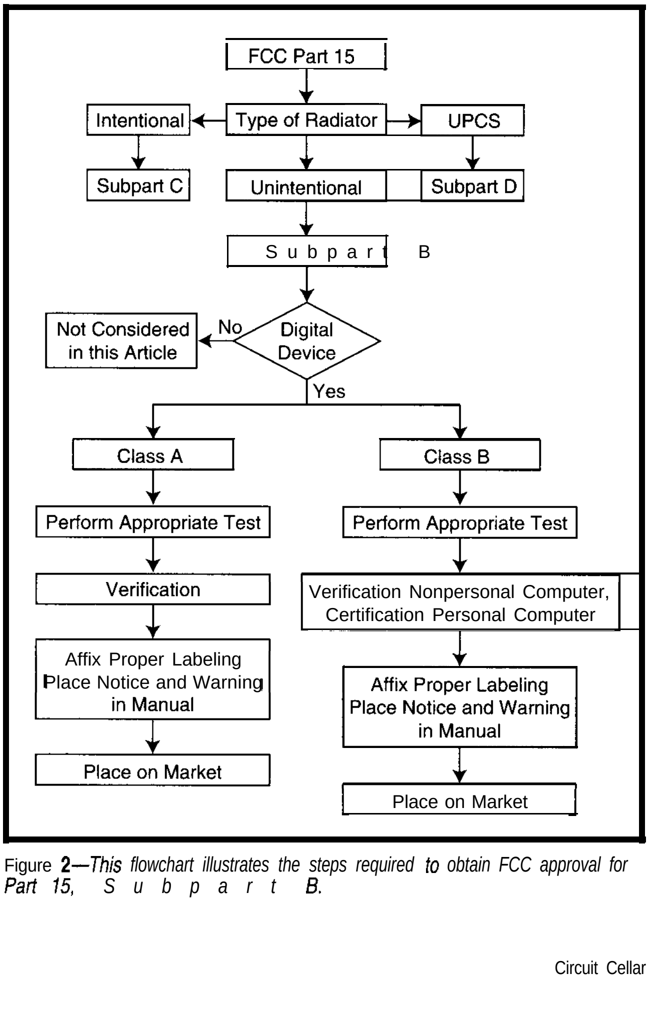


Figure 2—*This flowchart illustrates the steps required to obtain FCC approval for Part 15, Subpart B.*

produces radiated and conducted emissions as an byproduct of its operation.

Switching currents on wires and traces radiates RF. The faster the clock rate, the higher the frequency of the radiated RF. And, the faster the rise times of the switching currents, the fuller the frequency spectrum. Recall a dirac function has zero rise time and infinite frequency content.

Digital devices used in transportation vehicles
Power utility or industrial-plant electronic control systems
ISM (Industrial, Scientific, Medical) equipment (covered in Part 18)
Digital devices used in household appliances
Specialized medical devices used by health-care practitioners (covered by FDA)
Digital devices that consume less than 6 nW of power
Joysticks, mice, and other passive devices
Battery-operated devices that generate frequencies 4.705 MHz and that do not operate while connected to the AC mains

Table I--The FCC grants exemptions for some classes *of products for several reasons. Sometimes, the* product is *covered by another part of the code (i.e., ISM) or if may be exempt due to* the nature of *the product or its operating environment.*

Unintentional radiators (device or system) containing digital circuits are classified as digital devices (previously called computing devices). They generate and use timing signals or pulses at a rate in excess of 9000 pulses (cycles) per second and use digital techniques. The definition includes not only computers and microprocessors but digital logic such as PALS and gate arrays, as long as they operate at 9 kHz or more.

Part 15 subdivides digital devices into two categories-classes A and B. A Class A digital device is "marketed for use in a commercial, industrial or business environment, exclusive of a device which is marketed for use by the general public or is intended to be used in the home."

In contrast, a Class B digital device is "marketed for use in a residential environment notwithstanding use in commercial, business, and industrial environments."

For example, a Class A digital device is any piece of test equipment (e.g., spectrum analyzers) while a Class B example is a personal computer.

For FCC approval as a digital device, a device must undergo two tests-a radiated-emissions test and a conducted-emissions test. The standard that provides the basis for the tests prescribed by Parts 15 and 18 is ANSI 63.4-1992.

RADIATED EMISSIONS

The FCC limits the radiated-emission levels that an unintentional radiator can emit under Subpart B. The test procedure consists of the equipment under test (EUT) being placed at a distance, normally 330 m from a receiving antenna as shown in Figure la.

The EUT is operated normally, and the received radiation over the frequency band 30-1000 MHz is measured. In some cases, the upper frequency limit can be up to 40 GHz.

If the measured values are lower than the defined levels, the EUT passes. I'll provide more details about this test in Part 2 of this series.

CONDUCTED EMISSIONS

Most of us have experienced conducted emission (i.e., noise on an AC line caused by other equipment connected to the same line). To keep AC lines noise free, the FCC limits the amount of noise a product can send back on the line.

The basic test setup is as in Figure lb. The noise level that the EUT conducts back on the AC line is measured. The frequency band tested is from 0.45 to 30 MHz. As long as the EUT produces noise levels less than the limits set out in Part 15 Subpart B, it passes.

The radiated and conducted emissions tests for Class A and B digital devices are the same. However, Class A digital devices are allowed higher levels of radiated and conducted emissions.

Class A and B digital devices that aren't personal computers or peripherals are subject to verification. But, other Class B devices (i.e., computers and peripheral devices) are subject to certification.

With verification, the manufacturer must be satisfied that the device meets the appropriate standards prior to marketing. For certification, the FCC must be satisfied that the product meets the appropriate standards before marketing.

EMBEDDED BIOS™
BIOS Adaptation Kit
Just a few of our satisfied customers!
"We're impressed by the level of documentation and particularly by the readability of the code"
- M. Ryan, Cordant, Inc.
"Fast, flexible, high-quality code, and excellent technical support."
- L. Allen, Andros, Inc.
"Personally, I found the Adaptation Kit and tool set very straightforward to use, making the BIOS development process relatively easy."
- S. Chaplin, Software Engineer.
Step up to Embedded BIOS
Includes our award-winning Run-From-Rom DOS
Includes Flash Disk for popular Flash parts
Over 300 easy configuration options
BIDS Kit includes full source code
Royalties - $4/copy & down
Using one of these BIOS vendors? Switch over to Embedded BIOS for FREE*! Regular Price $4995
Phoenix, Award, AMI, SystemSoft
*Proof of existing license. Minimum 1000 units commitment. All TM names are property of their owners. Limited time offer.
For more info., Call 1-800-850-5755
General Software™ FREE-DEMO
320 108th Ave. N.E., Suite 400 • Bellevue, WA 98004
Tel: 206.454.5755 . Fax: 206.454.5744 . Sales: 800.850.5755
http://www.gensw.com/general • E-Mail: general@gensw.com
© 1996 General Software Inc General Software the GS Logo Embedded BIOS and Embedded DOS is a trademark of General Software All rights reserved
#118

ISM Freauencv	Tolerance
6.78 MHz	+15.0 kHz
13.56 MHz	+7.0 kHz
27.12 MHz	+163.0 kHz
40.68 MHz	+20.0 kHz
915 MHz	+13.0 MHz
2450 MHz	+50.0 MHz
6800 MHz	+75.0 MHz
24,125 MHz	+125.0 MHz
81.25 GHz	+250.0 MHz
122.50 GHz	+500.0 MHz
245.00 GHz	+1.O GHz

Table **2**—*Industrial, scientific,* and medical equipment can *have* unlimited radiating *power in these frequency bands, but the signal bandwidth is quite narrow.*

With verification, the manufacturer has measurements made to verify that equipment complies with the appropriate standards. Copies of the verification tests are kept at the production site.

The FCC label is placed on the device, FCC notices and warnings are placed in the owner's manual, and the device goes on market. At its discretion, the FCC can request a unit for its own tests (not a normal occurrence).

In contrast, with certification, after taking the same compliance measurements, the manufacturer makes an application to the FCC that includes a set of test results. The FCC reviews the results, maintains them in its files, and issues a certificate of approval.

At its discretion, the FCC can request a unit for its own tests. Before the device goes to market, the FCC label is placed on the device, and FCC notices and warnings are placed in the owner's manual.

Both types of devices must carry the FCC label. If it's impractical for the label to appear on the device (e.g., the device is too small), it may be placed in the operations manual or the container the device is sold in.

After approval, operation is subject to two conditions. It may not cause harmful interference. And, it must accept any interference received, including that causing undesired operation.

The first condition ensures that, even though the product has FCC approval, it cannot cause harmful interference. So, should you use your computer at an airport and it interferes with air-traffic frequencies, the FCC can stop you from using the computer even though it has FCC approval.

The second condition points out that the product must be designed with good engineering practices that let it operate in the EM environment set up by the radiated- and conducted-emissions tests. There are no FCC tests for EM1 susceptibility or immunity.

Certified products must bear an FCC identifier:

FCC ID :XXXnnnnnnnnnn

where XXX is the FCC grantee code assigned by the FCC and **nnnnnnnnnnn** is the Equipment Product Code.

The digital devices listed in Table 1 are exempt from FCC Part 15 Subpart B. Even though they're exempt, they're still subject to some general Part 15 rules. Most importantly, they cannot cause harmful interference.

These exemptions are one reason the European EMI/EMC standards that came into effect January 1, **1995 were** such a concern. Most of these exemptions do not apply in Europe.

Here, digital devices in household appliances are not tested under FCC regulations. However, they are subject to testing for the European market.

I'm not implying that products with FCC exemptions are not tested for EM1 or EMC. Exempt devices are tested by the manufacturer, often to levels more stringent than the new European tests.

But, Part 15 Subpart B does not normally apply to equipment that contains digital circuits covered by any other FCC regulation or is an intentional radiator. Therefore, you cannot make the case that your l-MW TV broadcast transmitter should be classified under Part 15 Subpart B because it contains digital circuitry. Figure 2 is a general flowchart for digital devices subject to Part 15 Subpart B.

FCC PART 18

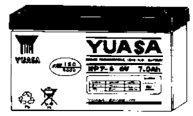
Normally, an intentional radiator is designed for broadcast or communications. But, one class of instruments intentionally radiates RF energy for functions other than communications.

These instruments range from RF welders to meteorological instruments. While they usually contain digital circuits, they're not covered by Part 15.

ALL ELECTRONICS CORPORATION

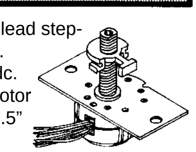
RECHARGEABLE SEALED LEAD ACID BATTERIES

6 Volts 7 A/h
Yuasa # NP7-6
5.95" X 1.3" X 3.7".
0.1875" qc connectors
$15.00 each
CAT# GC-67



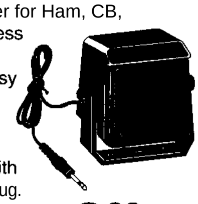
SMALL STEPPER MOTOR

Unipolar, two phase, 6 lead stepper motor. 53 ohm coil. Operates well on 12 Vdc. 15 degree per step. Motor body is 1" diameter X 0.5" 0.08" (2mm) diameter shafl is 0.365" long. The shaft is covered with a threaded plastic sheath (0.25" dia X 0.75" long) which acts as a worm gear. This mechanism, if not required, is easily removable. Rectangular mounting bracket, 1.73" X 1 .10".
$1.50 each
CAT# SMT-21
10 for $12.50

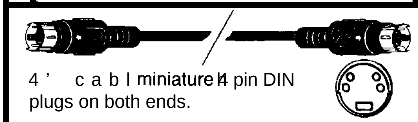


COMMUNICATIONS SPEAKER

This extension speaker for Ham, CB, marine radio or business communications improves clarity in noisy environments. Black case, grill and adjustable mounting bracket. 6 foot cord with 3.5 mm mini-phone plug. 2.78" X 2.6" X 2.1" deep.
CAT# SH-3
$6.00 each



S-VHS VIDEO CORD



4' cable miniature 4 pin DIN plugs on both ends.
CAT# CB-363
$3.00 each
10 for $25.00
100 for $150.00

ORDER TOLL FREE
1-800-826-5432
CHARGE ORDERS to Visa, Mastercard, American Express or Discover

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

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

MAIL ORDERS TO:
ALL ELECTRONICS CORPORATION
P.O. Box 567
Van Nuys, CA 9140
FAX (818)781-2653

E-Mail allcorp@allcorp.com
Internet - http://www.allcorp.com

#120

CIRCUIT CELLAR®

BACK ISSUES

1988
#6 Data Acquisition

1989
#7 Computing in Real Time
#8 Creative Computing
#10 The Electronic House
#11 32-Bit
#12

#13 Analog Signal Processing
#16 Communications
#18 Control Software

1991
#20 Portable Applications
Autorouters
#22 Embedded Applications
Emulators & Simulators

1992
#25 Building Automation
Embedded Signal Conditioning
#27 Real-Time Programming
Embedded Sensors & Storage
#28 Signal Processing
Embedded Interfacing
#29 Measurement 8 Control
Embedded Graphics &Video
#30 Debugging, Emulators & Simulators
Embedded Control 8 Conversion

1993
#31 Home and Building Automation
#32 Embedded Interfacing
#33 Data Acquisition
#34 Graphics and Video
#35 Communications
#36 Real-Time Programming
#37 Measurement 8 Control
#39 Power Control and Conversion
#40 Programmable Devices
#41 Embedded Control

1994
#42 Home Automotion
#45 Graphics &Video
#46 Robotics
#47 Distributed Control
#48 Communications
#49 Data Acquisition
#50 Embedded Applications
#51 Industrial Control
#52 Digital Signal Processing
#53 Computing in Real Time

1995
#54 Emulators & Simulators
#55 Embedded Interfacing
#56 Fuzzy logic
#57 Embedded Programming
Home Automation & Building Control
#58 Communications
#59 Signal Conditioning
#60 Graphics & Video
Home Automation & Building Control
#61 Digital Signal Processing
#62 Embedded Applications
Embedded PC
#63 Robotics
Home Automation & Building Control
#64 Analog Design
#65 Programmable Devices
Embedded PC

1996
#66 Power Supply Design
#67 Embedded Applications
Embedded PC
#68 Industrial Design
Home Automation & Building Control
#69 Communications
Embedded PC
#70 Cross-Development Tools
#71 Data Acquisition
Embedded PC
#72 Debugging Techniques
#73 Robotics
Embedded PC
#74 Embedded Programming
Home Automation &Building Control
#75 Fuzzy logic
Embedded PC
#76 Digital Signal Processing
#77 Graphics & Video
Embedded PC

Photocopies of SOLD OUT ISSUES are available for $4.00 each plus shipping.

#1 Inside the Box Still Counts
#2 Techies
#3 Control Magic
#4 Stepper Motors
#5 Remote Control Surveillance
#9 Intelligent Applications
#14 Applications in light
#15 Robotics
Building Automation
#17 Graphics
Embedded Applications
#19 Digital Signal Processing
Compilers for the 8051
#21 Communications
Home Automation
#23 Graphics&Video
Schematic Capture Software
#24 Applications in Light
Single-board Solutions
#26 Distributed Control
Embedded Programming
#38 Signal Processing
#43 Applications in light
#44 Cross-Development Tools

Send your requests to:
CIRCUIT CELLAR BACK ISSUES
4 Park Street • Vernon, CT 06066
Phone: (860) 875-2199 • Fax: (860) 871-0411

U.S. $4 plus $1 shipping per issue • Can/Mex $4 plus $2.50 shipping (airmail) per issue • Foreign $4 plus $4 shipping (airmail) per issue • All back issue orders require prepayment. VISA, MasterCard, or check accepted. All checks are to be payable in U.S. funds drawn on a U.S. bank.

The reasoning is quite simple-these devices require RF levels that are orders of magnitude greater than what's allowed under Part 15. The FCC has a special class for these instruments—CFR 47 Part 18 for ISM devices.

Some recognized ISM categories include industrial heating, medical diathermy, ultrasonic equipment, consumer ISM equipment, and magnetic-resonance equipment.

The FCC gave ISM devices their own set of frequencies as shown in Table 2. In the allotted bands, ISM equipment can have unlimited radiated power. The major source of unwanted EM1 is from digital circuits.

Outside the allocated frequency bands, ISM equipment is subject to both radiated- and conducted-emissions tests. The test setup is similar to that of digital devices under Part 15 Subpart B (see Figure 1).

The FCC distinguishes between ISM equipment intended for consumer and nonconsumer markets. Consumer ISM equipment is certified, while nonconsumer equipment is verified.

Those are the basic FCC standards and tests for products with digital circuits. Let's turn to the European standards.

EUROPE

When the countries of the European Community (EC) merged into one market, unified standards were one major issue to be resolved. Prior to EC, each country had its own set of standards.

If manufacturers didn't meet the standards of the country they were exporting to, their product was denied access. Clearly, this system was impractical under a unified market, since member states could use standards as nontariff trade barriers.

In an effort to unify standards, the EC passed a massive array of directives. Several directives can apply to a product, and of those, the Electromagnetic Comparability (EMC) Directive is the most onerous.

The European EMC Directive 89/336/EMC was adopted on May 3, 1989, amended by Directive 92/31/EMC, and went into effect on January 1, 1996. It applies to any apparatus that's liable to cause or be affected by EMI.

99 MHz 8051 SBC!

You've heard about the Dallas 80C320, and 805 1 compatible CPU. It can really get the job done fast, but you need *to get started right now.* The 320SBC features the '320 on a board that is ready to go -- NOW.

HTE's 32OSBC features:

- High speed 805 I Instruction set, executes instructions up to 3X as fast as a standard 8051 at the same crystal speed.
- Cost effective: $179 for the -50 development version, $149 for the -10 OEM version in single qty.
- Two serial ports: RS-232 & 5V
- Optional 3rd RS-232 serial port can be used to free both the '320 serial ports for your application.
- On board monitor eliminates the need for an ICE in most cases.
- Development tools for C or ASM
- Production ready design - Now!
- Special configurations available that will reduce cost and meet your specific needs by including only the features you need.
- Compact size fits anywhere!

HTE

HiTech** Equipment **Corporation
(619) 566-1892 • Fax: (619) 530-1458
info@hte.com http://www.hte.com

#122

	European	FCC
Product-specific Emissions Standards		
ISM	EN 55011	CFR 47, Part 18
Radio/TV	EN 55013	CFR 47, Part 15
Household Appliances	EN 55014	Exempt
Luminaries	EN 55015	Exempt
Information Technology	EN 55022	CFR 47, Part 15
Product-specific Immunity Standards		
Radio/TV	EN 55020	N/A
Generic Emissions Standards		
Generic Class;Residential, Commercial, and Light Industry	EN 50081-l	N/A
Generic Class;Industrial	EN50081 -2	N/A
Generic Immunity Standards		
Generic Class;Residential, Commercial, and Light Industry	EN 50082-l	N/A
Generic Class;Industrial	EN 50082-2	N/A

Table 3—*This* fable lists European product-specific and generic test standards *and* their FCC *equivalents. The FCC has no immunity standards and several exempt products.*

Its Article 4 requires that apparatus be constructed so:

- the EM disturbance generated doesn't exceed a level where radio and telecommunications equipment and other apparatus operate as intended
- it has an adequate level of intrinsic immunity to EM disturbance to enable it to operate as intended

Here, the FCC and European standards significantly differ. Although the FCC rules state that equipment must tolerate interference, it doesn't mandate immunity tests. However, in Europe, a product must pass tests on both emissions and immunity standards.

You can comply through self-certification or type approval. Type approval requires a technical construction file and the services of a European Competent Body (ECB) (i.e., an EC-accredited consultant). Type approval is for specific cases and is used much less often, so I won't discuss it further.

SELF-CERTIFICATION

Self-certification offers great flexibility. Within reason, the manufacturer can choose the standards that compliance is sought under.

There's no need to engage an outside test house or-in fact-that tests actually be performed. Legally, all that's required is a declaration by the manufacturer or its European agent that the equipment complies with the EMC directive. But of course, performing no tests subjects the company to a potentially bankrupting liability should an EM1 problem arise.

Self-certification has five steps. First, the manufacturer or testing lab tests to ensure the equipment complies with the appropriate standards.

Next, a technical file is written to show how the product complies with relevant standards. It includes a rationale for selecting the standards as well as the results of the compliance tests.

A Declaration of Conformity describes the equipment (with model and serial numbers) and the directives and standards that define conformity. It must be *signed by a person empowered to bind the company.*

The CE mark is affixed to the equipment, packaging, manual, and/or instructions. And finally, the product enters the market. The manufacturer must ensure that 80% of all of the product must meet the EMC directive to an 80% confidence level.

STANDARDS

The Europeans are taking a different and more complex route to standards development. They are developing product-specific standards.

Each product-specific standard is general in nature and calls other standards to provide compliance details.

When no product-specific standards exist, generic standards based on the environment the product is used in are applied. Table 3 lists some product-specific and generic standards and the FCC equivalents.

The basic standards and tests methods are developed by the International Electrotechnical Commission (IEC) and by the Comité Internationale Speciale des Perturbations Radioéléctriques

IEC Standards	
1000-4-2	Electrostatic Discharge-Range: 2-l 5 kV (8-kV contact)
1000-4-3	Radiated RF immunity-Range: 1, 3, 10 V/m, 80-1000 MHz, 80% AM modulated
1000-4-4	Electrical fast transients-Range: 1 kV on AC lines, 500 V on I/O and DC lines
1000-4-5	Surge-Range: 0.5, 1, 2, and 4 kV
1000-4-6	Conducted disturbances' Induced by radio-frequency fields-Range: 1, 3, 10 V EMF unmodulated
1000-4-8	Power frequency magnetic field-Range: continuous 1, 3, 10, 30, or 100 A/m; short duration 300 or 1000 A/m
1000-4-9	Pulse magnetic field-Range: 100, 300, and 1000 A/m
1000-4-l 0	Damped oscillatory magnetic field-Range: 10, 30, and 100 A/m
1000-4-l 1	Voltage dips, short interruptions, and voltage variations
1000-4-l 2	Oscillatory waves
CISPR Standards	
11	Limits and method of measurement of radio interference characteristics for industrial, scientific, and medical (ISM) radio-frequency equipment, fluorescent lamps, and luminaries
16	Specifications for radio-interference measuring apparatus and measurement methods
22	Limits and method of measurement of radio-interference characteristics of information technology equipment (ITE)

Table 4—*IEC and CISPR standards are* ***"technical"*** *standards (i.e., they lay out the technical details of the EMI tests). The calling standard* ***can*** *therefore be general in nature, facilitating standards development.*

(CISPR). The IEC and CISPR standards relevant to this series are in Table 4.

The IEC details the required tests, while the CISPR provides test methods. This greatly simplifies the process of developing a new product standard.

The new product standards need only to lay out the required levels and tests. The details of the tests and test methods are in place via the IEC and the CISPR.

For the equipment I'm discussing (i.e., equipment with digital circuits that are the major source of EM emission), generic standards apply. Which standard applies depends on the environment the equipment is used in.

For example, if the environment is residential, commercial, or light industry, the generic standards EN 5008 1- 1 and EN 50082-l are likely the most applicable. The applicable tests are:

- emissions-Radiated and Conducted emissions based on EN 50022 (similar to FCC Part 15 for digital devices]
- immunity-IEC 1000-4-2: Electrostatic discharge; IEC 1000-4-3: Radiated RF immunity; IEC 1000-4-4: Electrical fast transients

As you can see, the European standards are much more involved than the FCC rules.

COMING UP

In the rest of this series, I look at the technical aspects of emissions and immunity tests. I deal with emissions standards in Part 2 and immunity in Part 3. Part 4 covers testing labs.

Keep reading. You'll gain an understanding of how to design for EMI/EMC. By handling EMI/EMC issues at the design stage, you can save both time and money, as well as enhance your product's performance. ❑

foe 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 Joe 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

413 Very Useful
414 Moderately Useful
415 Not Useful

#123

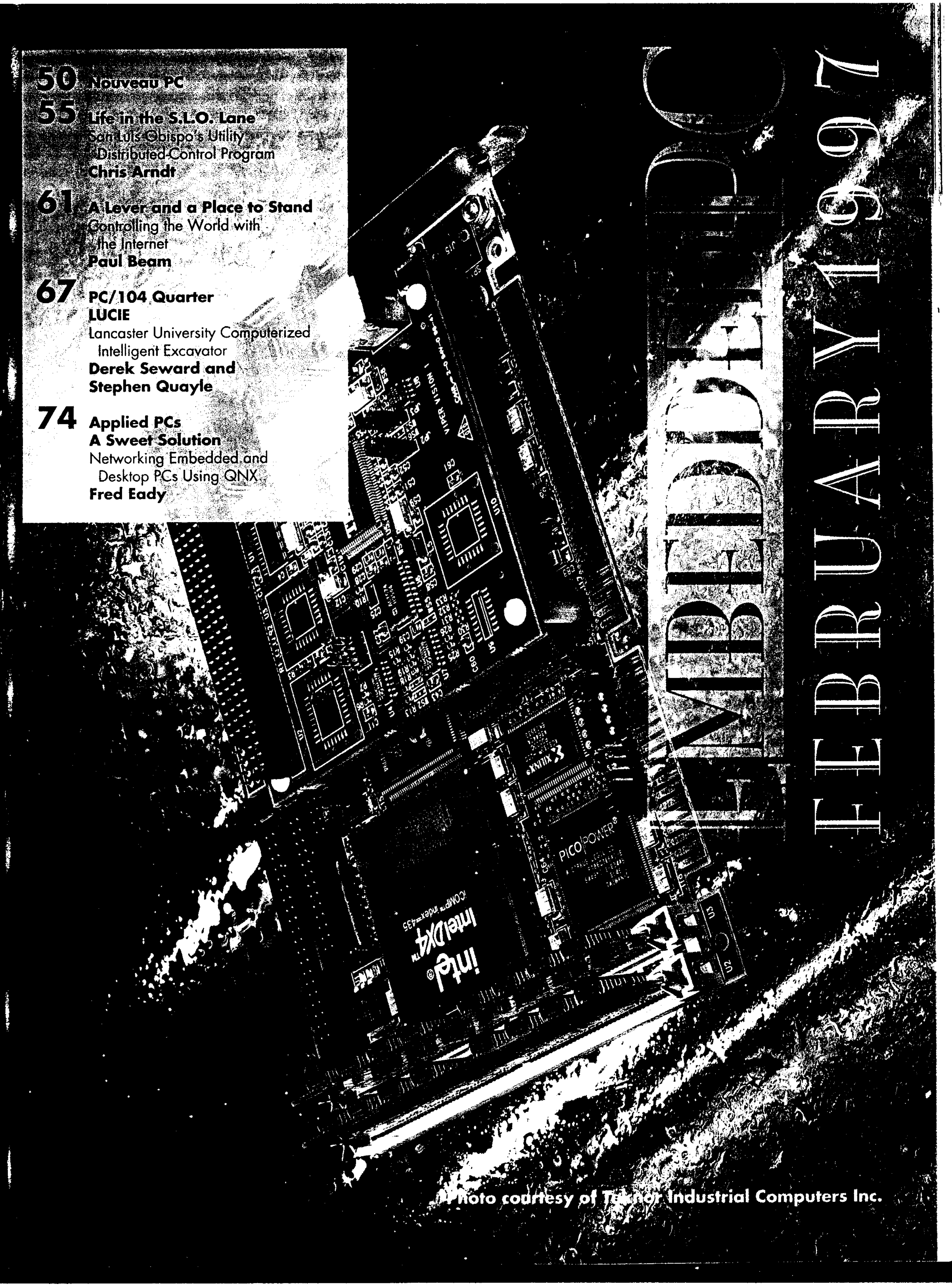
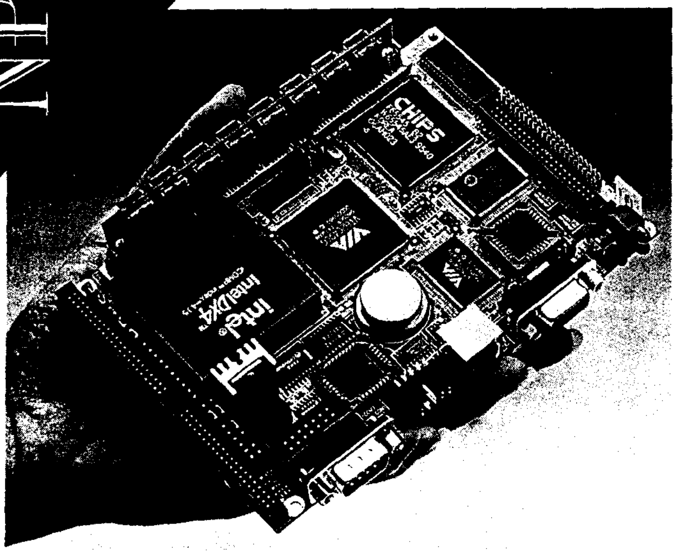


Photo courtesy of Teknor Industrial Computers Inc.



COMPACT '486-BASED PC

The **Biscuit PC** series is a cost-effective solution for limited-space applications (e.g., medical computing devices, machinery, point-of-sales, and gambling control). Based on PC/AT architecture, these computers are compatible with most off-the-shelf software and can significantly reduce development time and cost. They pack the features and processing power of a standard industrial computer into the size of a 3.5" hard drive (4" x 5.7").

Major features include a PCI-bus SVGA controller with LCD display, PCI-bus EIDE controller, and 16-bit PCI-bus Ethernet interface, which is Novell NE 2000 compatible. Additional hardware integrated into the board includes two high-speed serial ports, one multimode parallel port, fourchannel DIO, a mini-DIN PS/2 keyboard, and a PS/2 mouse interface. Power-saving modes for I/O peripheral devices are also supported.

Prices for the Biscuit PC start at $166.

American Advantech
750 E. Arques Ave.
Sunnyvale, CA 94086
(408) 245-6678
Fax: (408) 245-8268

#510

EMBEDDED I/O MODULES

Arcom Control Systems introduces five PC/104 industrial expansion modules that offer economic combinations of digital or serial I/O. **The AIM104** series can be used with any PC/104-compatible board. With their onboard signal conditioning or interface to the industry-standard Signal-Conditioning System, the modules are aimed at machinery and automation-system builders seeking cost-effective control and data-acquisition systems.

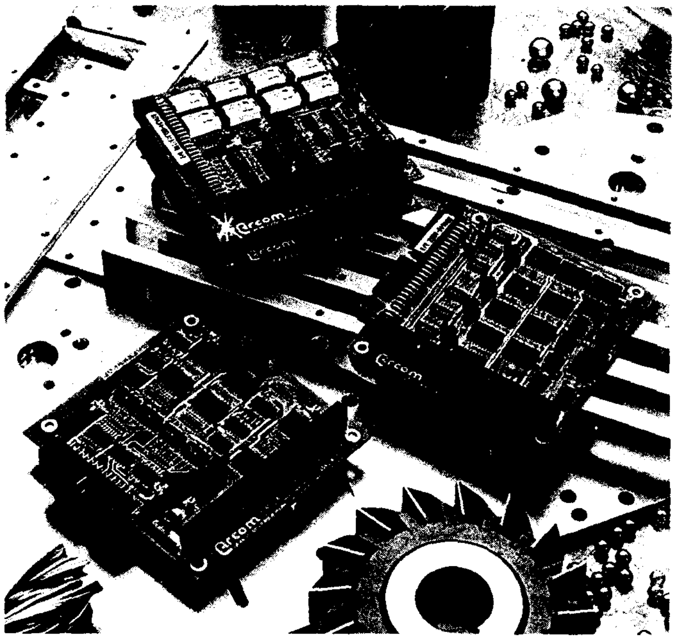
Three digital functions come with onboard signal conditioning and provide a choice of 16 channels of optoisolated inputs with link-selectable debouncing, 16 channels of optoisolated outputs, or a combination of 8 optoisolated inputs and 8 relay outputs rated to 60 W. The relay module lets you define the power-up state of the outputs for fail-safe operation.

Another digital module offers 32 channels of bit-programmable I/O that defines the power-up state in banks of eight. The last module offers four programmable serial communications channels with asynchronous transmission at rates to 57.4 kbps or synchronous transmission at up to 1.9 Mbps. Based on the Zilog 85230, it includes deep FIFO buffers to optimize data throughput. On both modules, I/O is routed via a 50-way ribbon cable connector with a pinout conforming to the Signal-Conditioning System.

All five modules are supplied with a driver library and C *source* code. Prices range from $135 to **$180.**

Arcom Control Systems, Inc.
13510 S. Oak St. • Kansas City, MO 64145
(816) 941-7025 • Fax: (816) 941-0343

#511



Nouveau-PC

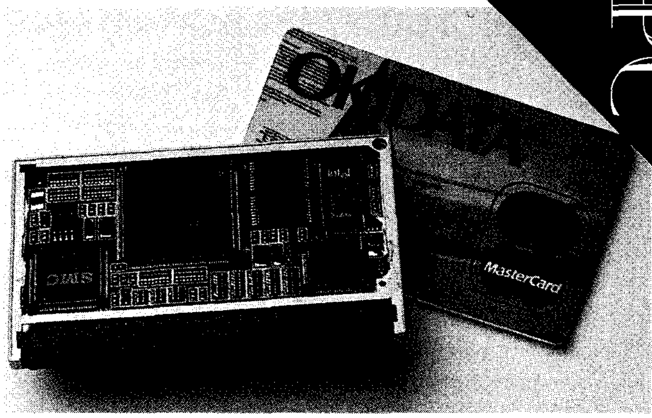
edited by Harv Weiner

CREDIT-CARD-SIZED '486 PC

The **comPCard 486-50** and **486- 100** are i486-based PC modules capable of 50 and 100 MHz, respectively. Both modules are fully functional IBM PC/AT motherboards packaged in the size of a credit card (measuring approximately 3.4" x 2.2"). The cards are designed for endurance with guaranteed shock resistance of up to 100 G and vibration resistance up to 15 G.

Through its 236-pin EASI interface connector, the comPCard '486 family supports an ISA bus, serial and parallel ports, mouse, keyboard, and IrDA SIR (infrared interface). Also included is a speaker, floppy disk drive, IDE hard disk drive, CRT and LCD interfaces for three types of flat-panel displays (TFT/D-STN/STN), and power management. The card family also supports PCMCIA-ATA boot functions. A PCMCIA hard disk drive can be used as the system disk to reduce the overall size of the system. An SVGA graphics accelerator with 5 12 KB of VRAM connects to an internal VLB to offer high-performance graphics capabilities.

The comPCard family can be configured with a 1-, 2-, 4-, 8-, or 16-MB memory module. Future expansion through the use of Small Outline Dual Inline Memory Modules (SO DIMM) is possible. By using a 3.3-V power supply, along with power-saving and suspend/resume functions, overall power demand in the comPCard 486-50 reduces to -1.5 W. The card also features very low power consumption in suspend mode (60 mW).



Okidata
532 Fellowship Rd. • Mount laurel, NJ 08054
(609) 235-2600 • Fax: (609) 778-4184

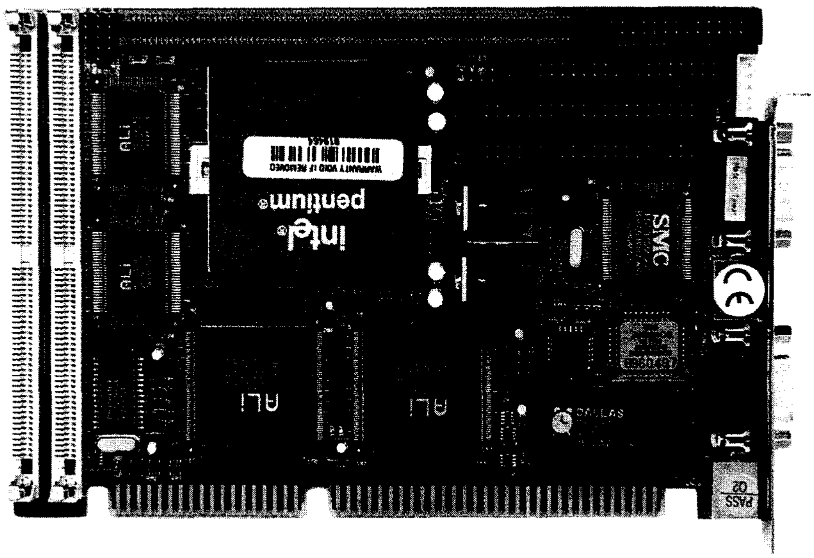
#512

6x86 EMBEDDED PC

The **IPH-686** half-size ISA Pentium board contains all the basics of a standard IBM PC/AT-compatible desktop computer plus some unique features that make it ideal for high-speed embedded applications. The IPH-686/166 MHz performs at a Landmark V.2.0 rating of 964 MHz and supports 256-KB through 1 -MB pipeline burst and standard cache RAM modules for high-speed access to data blocks.

The board includes a Super AT peripheral chip to provide two high-performance 16C550-compatible serial ports, an ECP/EPP-compatible bidirectional parallel port, a dual floppy-disk port, and two EIDE hard-disk ports. Also included are a PS/2 keyboard port, a PS/2-compatible mouse port, an onboard speaker, watchdog timer, clock/calendar with integral lithium battery, and up to 64 MB of DRAM. Since the IPH-686 was designed for embedded and industrial applications, the BIOS permits booting without a keyboard or monitor.

The watchdog timer makes the board well-suited for controlling critical processes where unattended operation is essential. In the event of an I/O timeout delay or external failure, you can program the watchdog timer to generate a non-maskable interrupt or system reset. The timeout delay is adjustable from 1 to **220 s.**



The 100-MHz IPH-686 SBC comes complete with a user manual and a two-year warranty. It is priced from $875.

Micro Computer Specialists, Inc.
2598 Fortune Way • Vista, CA 92083
(6 19) 598-2 177 • Fax: (6 19) 598-2450

#513

Nouveau PC

FLAT-PANEL DISPLAY SBC

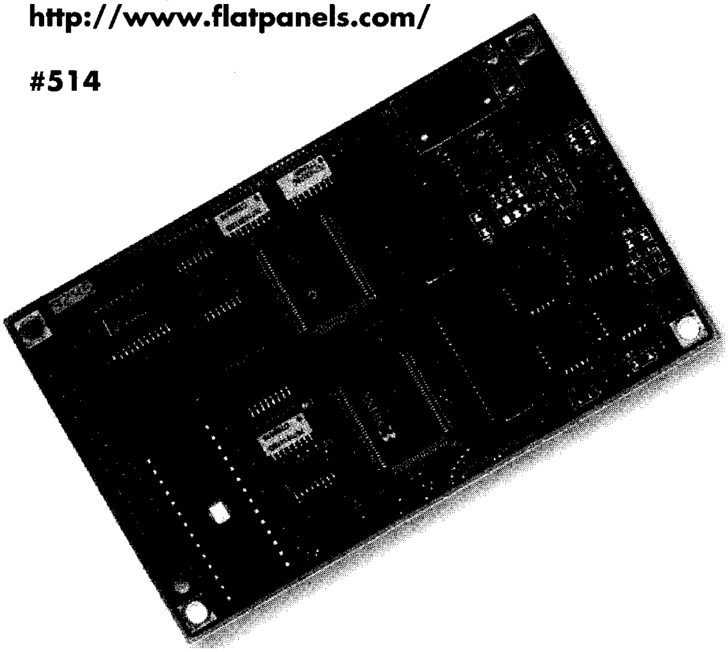
The Pixel Press is a highly integrated, compact single-board computer that combines the power of a 32-bit ARM RISC processor, I/O support (serial, parallel, Ethernet, and SCSI), and the flexibility of reconfigurable Xilinx FPGA and PAL programmable logic into a low-power module that drives any flat-panel display with resolutions up to 1280x 1024. The board can beeasilyconfigured for applications as diverse as a VGA 6" monochrome LCD flat panel with RS-232 input or a 17" flat-panel display with 1280 x 1024 resolution and Ethernet TCP/IP network input. No disk or bus interface is required.

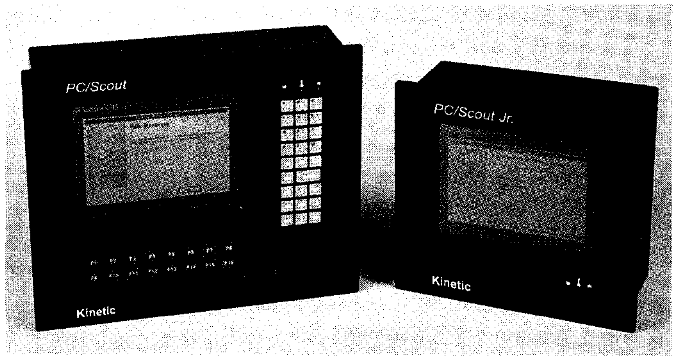
A unique feature of the Pixel Press is its PCX graphics-file display capability. PC-generated PCX images, which previously had to be redrawn every time, can now be stored and displayed. This capability increases productivity, reduces processor requirements, provides for quicker development, and improves data transfer.

The Pixel Press sells for under $250.

Applied Data Systems, Inc.
9140 Guilford Rd.
Columbia, MD 21046
(30 1) 490-4007
Fax: (301) 490-4582
http://www.flatpanels.com/

#514





RUGGEDIZED PC

PC/Scout is a modular, ruggedized, panel-mount PC system that is available with either PC/104 or half-length ISA expansion slots. The system includes a high-brightness color-LCD VGA display with self-adjusting contrast control and a resistive-touch front panel. The unit is intended for OEMs and systems integrators in factory automation, process control, environmental monitoring, and other applications that need PC technology in physically challenging situations.

These network-centric industrial-PC products work with peripherals, sensors, and instruments common in industrial applications. 1 OBaseT and 10Base5 Ethernet support is standard. Optional interfaces include DeviceNet, Interbus-S, and Proxim wireless Ethernet LANs. PLC support is also available via add-in expansion cards.

PC/Scout is available with either an Intel '486DX4-100 or a Pentium P166 CPU. Additional features include two serial ports (one configurable as an RS-485 industrial communications link), one parallel port, floppydiskdrive, 640-MB hard drive, and up to 32 MB of DRAM. Package features includea touchscreen, touchpad, membrane keyboard, and NEMA-4–compliant front panel. PC/Scout ships with your choice of Microsoft Windows 3.1, Windows 95, or Windows NT operating systems.

An 8-MB '486DX4-100 system with industrial touchscreen, Ethernet interface, two PC/I 04 slots, two serial ports, and floppy and 640-MB hard drives sells for $4495. OEM discounts are available.

Kinetic Computer Corp.
270 Third St.
Cambridge, MA 02 142
(617) 547-2424
Fax: (617) 547-7266

#515

FOUR-CHANNEL SERIAL INTERFACE

The **ZT 6660** is a 32-bit, 3U-size, CompactPCI serial interface board that features four serial channels, implemented with 16650 UARTs (8250 compatible), at speeds up to 1 15 kbps. With the D1 option, each channel is individually configurable for RS-232 or RS-485 operation. The D2 version provides RS-422/-485 optical isolation for electrically noisy environments. Each RS-422/-485 channel is isolated from the serial cable by 300 V. The maximum data rate with optical isolation is 56 kbps.

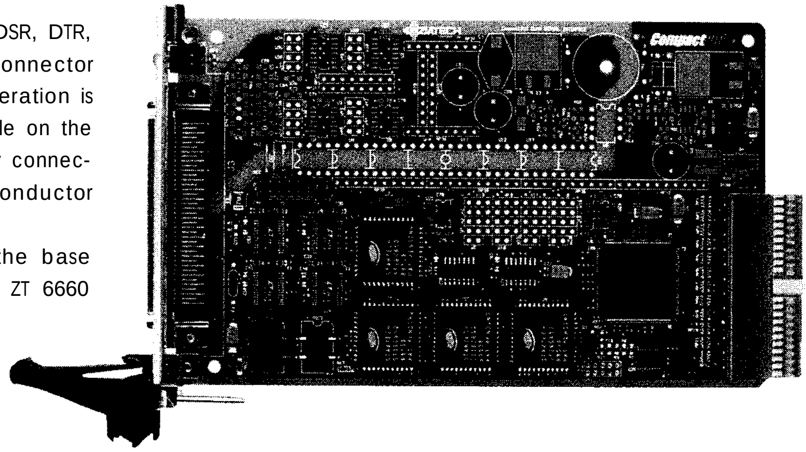
All RS-232 channels terminate TXD, RXD, CTS, RTS, DSR, DTR, RI, and DCD signals to half of an 80-pin SCSI-type connector configured for DTE operation. When RS-422/-485 operation is selected, transmitted and received data is made available on the other half of the connector. Both arrangements allow easy connection to a flat cable, 9-pin, D-type connector without conductor scrambling or soldering.

The ZT 6660 requires 64 I/O port addresses, with the base address being determined by the system BIOS. Both ZT 6660 versions include the PCI plug-and-play registers and an interrupt latch for software compatibility.

The ZT 6660 sells for $375.

Ziatech
1050 Southwood Dr.
San **Luis** Obispo, CA 93401
(805) 541-0488 • Fax: (805) **541-5088**
info@ziatech.com
http://www.ziatech.com/

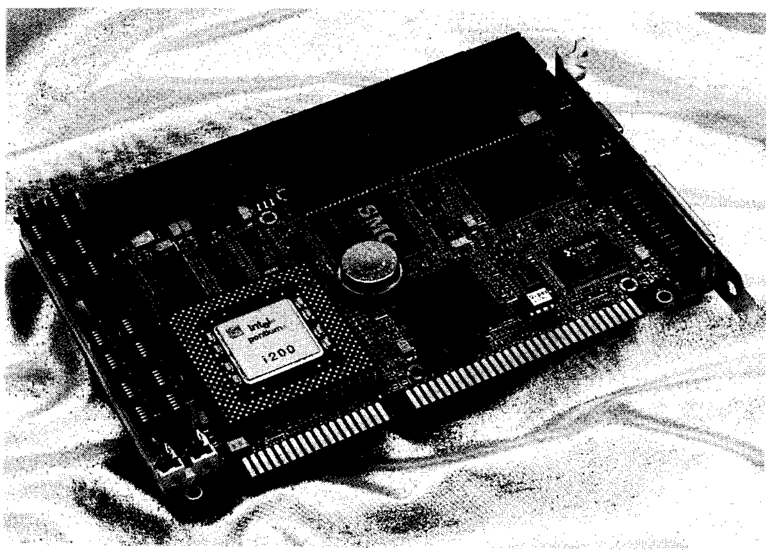
#516



200-MHz SBC

The VIPer820 is a half-size industrial single-board computer that integrates Pentium processors up to 200 MHz, PCI Fast SCSI II, and a complete range of I/O and advanced firmware. Designed **primarily** for high-speed portable systems, the SBC is also well-suited for embedded applications due to its ISA-bus connector, PC/ 104 interface, and stand-alone operation modes. It was developed with a plug-and-play-compatible philosophy for easy peripheral interfacing. And, it supports advanced power-management functions that reduce power consumption.

Built-in functions include boot-block flash BIOS, PCI EIDE hard-disk interface, and simultaneous flat-panel and CRT operation. It also has serial, parallel, keyboard, and mouse ports, as well as supervisor utilities (e.g., watchdog timer and power-failure detector). It supports up to 256 MB of DRAM, 256 KB of Level-2 cache, up to 4 MB of bootable flash memory, and 1 MB of video DRAM.



The VIPer820 with a 75-MHz Pentium processor, 1 -MB video DRAM, 256-KB synchronous cache, and a PCI Fast SCSI II (without memory) is $1569.

Teknor Industrial Computers, Inc.
7900 Glades Rd.
Boca Raton, FL 33434
(561) 883-6191
Fax: (56 1) 883-6690

#517

Nouveau-PC

Chris Arndt

Life in the S.L.O. Lane

San Luis Obispo's Utility Distributed-Control Program

Here's a practical hands-on example of embedded PCs doing their *thing. San Luis Obispo* uses embedded PCs to manage its water and wastewater utilities. Smart remote computers coordinate with a central system to monitor parameters.

In the old days, industrial process control was pretty easy. It was all done by hand.

If you wanted to fill a tank with water, you turned on a pump. When the tank got full, you shut it off. With a telescope, you could even do this at a fair distance, as long as it wasn't dark and you could see the indicator on the tank.

After a while, switches and floats automated the tank-filling process. The tank and the pump station had to be close together, or there was a lot of expense and maintenance in running control wires. The more functions the system had (e.g., lead and lag control for multiple pumps and low- or high-water alarms), the more conductors were required.

When distances got too long, a telephone-system infrastructure was built up. Control engineers figured out how to perform multiple control and alarm functions by sending various audio tones via the phone and decoding them into different functions at the far end. So, several sites were monitored and controlled remotely.

CONTROL AND CONTROLLERS

This system basically describes distributed control's older sibling, SCADA (Supervisory Control And Data Acquisition), which is illustrated in Figure 1. Later control engineers realized that, with all this field data brought back to one central location, they could take one new-fangled computer, wire in the field equipment, and run pretty elaborate control routines.

As electronics grew more sophisticated, hard-wired or tone-remote equipment was replaced by dumb remote field terminals that communicated with the central computer via modem and digitized signals sent to the supervisory computer. They also decoded return data into varying analog outputs or control relay closures.

However, the centralized supervisory computer relied on one computer for total system control. As long as it ran and the phone lines were intact, all the tanks were full of water and everyone was happy.

But, when the computer crashed or phone lines were knocked out, tanks were not kept full. No one was happy, and the water-distribution crew was reduced to manually controlling the system until things got repaired.

DISTRIBUTION OF CONTROL

Microprocessors enabled engineers to spread out the control-system intelligence and make it more fault tolerant. Small controllers at each location provided more sophistication than float-switch control. It wasn't as powerful, but it was more reliable.

Next step: combining the new, smarter, remote-control computers with SCADA's central-computer concept. That way, the smarts are in the field, the operator interface is in the shop, and the entire system can be monitored remotely.

That's the basis of utility's Distributed Control System (DCS). Smart, embedded microprocessor-based Distributed Process Controllers (DPCs) are in the field, communicating with one or more graphic man-machine interface (MMI) computers in central locations.

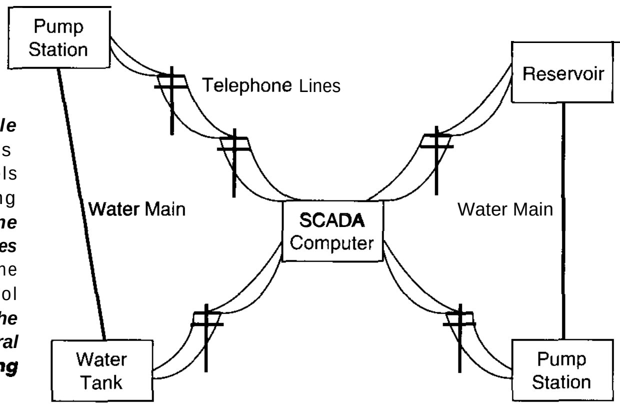


Figure 1: ***This simple SCADA system*** transmits tank and reservoir levels to a central controlling ***computer via telephone*** line;. Different ***phone lines*** send control signals to the pump stations to control the ***pumps based on the water levels. If the central computer fails, everything*** stops working.

let's take a detailed look at these DCSs. Even though this system is for a small municipal drinking-water and wastewater utility, the distributed-control concepts are easily extrapolated to other situations.

DIRTY TO CLEAN WATER

The City of San Luis Obispo, California, Utilities Department supplies drinking water and treats wastewater for -50,000 people. DCS is used in every facet of raw water supply.

In water supply, distributed control offers remote automatic and manual control of pump stations and turnouts or valves along an 18-mi. pipeline (see Figure 2). In the water and wastewater treatment plants, DPCs operate the treatment processes and log flows and other measurements.

In the wastewater-collection system shown in Figure 3, each lift station is operated and monitored by a DPC, and all operating data and alarm conditions are reported to the section shop. Alarms are sent via automatic voice dialer to a pager.

The city's equipment of choice is the Bristol Babcock Network 3000 line of controllers. The BBI 3330 is the predominant DPC in all our systems.

The 3330 in Photo 1 is similar in concept to the computer on your desk. Current models feature a '386 processor, while older ones have a '186. A typical controller has 64 or 128 KB of onboard RAM, and ROMs contain the OS kernel.

The backplane shown in Photo 2 contains edge connectors for inserting the processor, communications, and I/O cards, as well as a couple of general-purpose IBM-compatible slots. The I/O cards are single-purpose boards with multiple connections for analog inputs and outputs, digital (switch closure or voltage signal) inputs or outputs, or high-speed counters. Depending on the model, there are 0, 6, or 12 I/O cards on a backplane, and serial remote I/O racks can extend this count.

Each 3330 has an assortment of I/O cards and runs a unique control program written in Bristol Babcock's high-level ACCOL language. ACCOL is a compiled language with prebuilt modules for many processcontrol and communications needs.

The 3330s communicate over hardwire or radio in a master/slave tree network. Depending on address-space limitations, this network can be up to 7 levels deep and 128 nodes wide on each level.

It is possible for a 3330 to be simultaneously a slave and a master to different network parts as shown in Figure 2. The communications protocol is similar to InternetTCP/IPoramateur-radiopacketing.

Each node is assigned a unique numerical address I-I 27 on its level and a node name up to four characters long that identifies it to the network. When network topology files are created during network programming, a global address isassigned.

The distributed nature of this control system lends itself to the diverse nature of water and wastewater utilities. We program individual field units with the intelligence to run stand alone, but we also take advantage of networked communications.

To make this possible, we use several forms of data communication. In treatment plants, the data path is a 9600-bps RS-485 multidrop cable.

Unfortunately, RS-485 needs to be continuous run, which means lots of doubled-up runs if the buildings or DPC locations aren't linear. I get around this by using an opto-isolated RS-485-to-RS-485 repeater to make a spur off the multidrop cable.

Another disadvantage is the cable's length limit. We successfully used a single cable run of -1 mi. With newer isolators, repeaters, and fiber modems, RS-485 can run as far as your checkbook allows.

We recently lit up a 9600-bps fiber-optic link using fiber-to-RS-232 modems to connect a remote network.

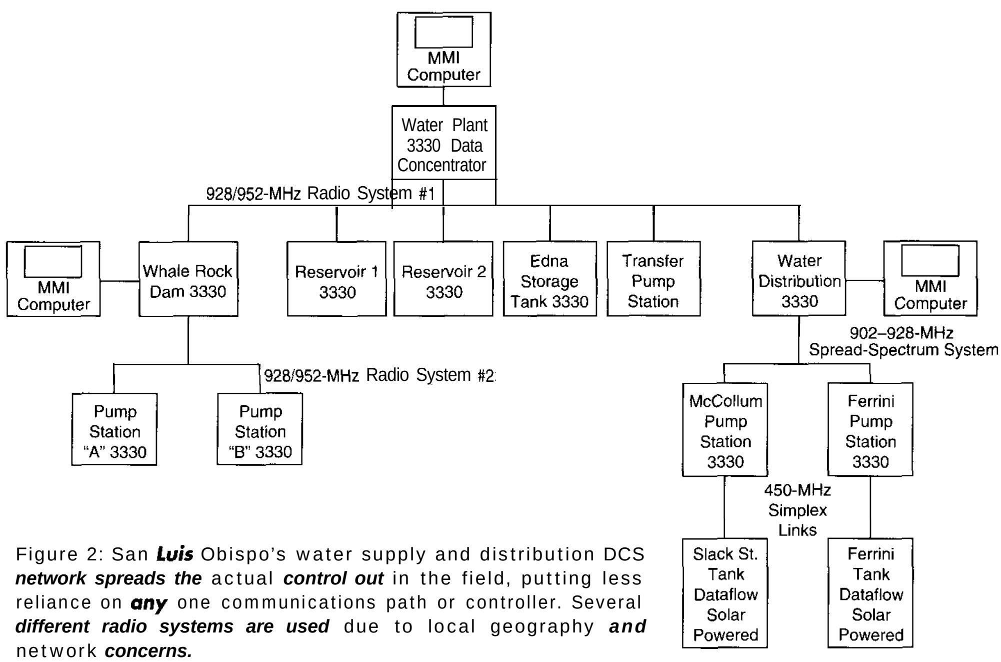


Figure 2: San ***Luis*** Obispo's water supply and distribution DCS ***network spreads the*** actual ***control out*** in the field, putting less reliance on ***any*** one communications path or controller. Several ***different radio systems are used*** due to local geography ***and*** network ***concerns.***

Fiber has one big advantage over copper-its insensitivity to electromagnetic fields. Before we figured out the surge suppression, lightning several milesaway induced enough EMF in that buried l-mile-long cable to blow the line driver and receiver chips.

In the field, we use 928-/952-MHz band-radio repeaters for the system's backbone. The oldest ones in service still use 1200-bps Bell 202 AFSK modems. We upgraded the wastewater-collection system repeater to use Microwave Data Systems (MDS) 9600-bps radio modems and data repeaters.

The upgrade cut the polling period (i.e., the time to interrogate and receive data from the nine in-field DPCs) from 45 to 8 s. If 45 s doesn't seem long, try making changes in an emergency. Time adds up.

RADIO DISTRIBUTEDCONTROL

The MDS radio system is a sort of a DCS of its own. By plugging a computer into the master station or repeater, every radio transceiver can be queried individually for variousoperating parameters (e.g., supply voltage, received signal strength, frequency deviation, forward and reflected power, and VSWR).

Not only can these be reported to the computer and logged in a file, but over-the-air adjustments of outputpower, deviation, and frequency can be made to remote radios.

Each radio contains a remote diagnostic card with EE pots in parallel with the regular mechanical potson the main circuit board. When remote diagnostics are installed, the mechanical pots are set at full range, so all adjustment comes from the EE pots.

The MDS radio system handles remote maintenance via preassigned addresses, called loopback codes, for each radio transceiver. The code is usually the last four digits of the radio's serial number.

This address is programmed into the radio's EEPROM. it's highly unlikely that two radios in the same system have the same loopback code. If that happened, we'd simply reprogram one radio's code.

All inquiries to field radios and responses to the master station are made via DTMF tones. DTMF is used rather than the data channel to avoid sending false or inaccurate data to any equipment connected to the radio transceivers. Also, in some cases, the communications system uses external modems, making them unavailable to the internal radio diagnostics.

The MDS23 10 radios in Photo 3 are computer-programmable frequency-synthesized 928-/952-MHz transceivers. The main board uses a 68k processor.

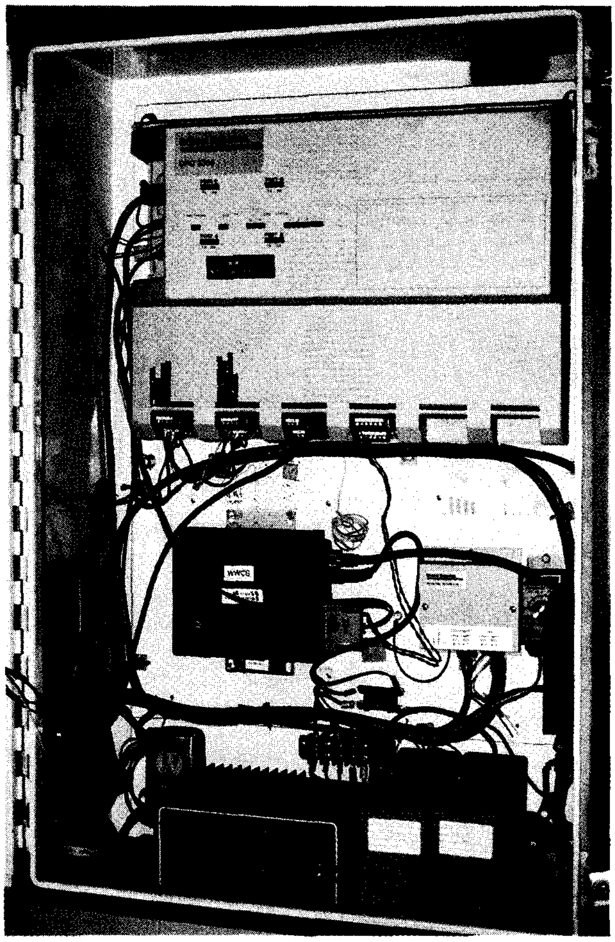


*Photo I: Below the **Bristol Babcock** 3330 Data Concentrator is an **MDS23** 10 radio, back-up batteries, and a **24-VDC power supply.** It communicates **with** remote units and feeds data **to** the **man-machine** interface (MMI) computer. **Its** programming drives a voice-synthesized alarm autodialer based on incoming data.*

The two slots on the motherboard are for a remote diagnostics card and a modem card. At least the 9600-bps modem and maybe one or both of the others have 68k microprocessors to handle the data flow.

WIRELESS CURRENT LOOPS

Early on, one of the difficult parts of the water-distribution telemetry system was getting a simple depth signal from a storage tank on a hill down to the pump station to control the pumps.

The first implementation of remote-level measurement used a complicated float- and cam-driven timed-voltage pulse sent from the tank to a similar cam arrangement at the pump station. If cable was installed, there was no problem, but sometimes the cable was damaged.

One vendor tried to solve this by measuring the total head or system pressure at the pump house (around 100 psi). The water depth in the 30' tall tank was the top 13 psi on that pressure, which was marginally OK when the pumps were off.

But when they ran; there was dynamic system pressure due to the pumps overcoming line friction to move water up to the tank. Thus, the turn-on pressure setting for the pumps was based on static system pressure, and the turn-off setting was based on dynamic pressure.

Tocomplicate matters, dynamic pressure changed depending on whether the small regular pump or the large fire-flow pump was running. To a lesser extent, the actual use in that pressure zone at the time also affected the pressure.

When this system became too hard to control, we replaced it with a DPC connected to a submersible 4–20-mA pressure transmitter, a DPC at the pump house to control the pumps, and a radio link using Bell 202 protocol modems to carry the digital signal.

Admittedly, this expensive solution was overkill. But, our options were limited, and it worked.

Today, there are single-purpose devices for this. We use a radio transmitter/receiver pair [not transceivers] that digitizes the analog signal, transmits it by AFSK over the radio, and converts it back to an analog 4–20-mA signal. It's essentially a wireless 4–20-mA current loop.

The Dataflows run on 12 VDC from an internal or external power supply, so they're easy to use remotely. After installing a solar panel, we add a charge regulator and battery. Both ends have address-setting switches, so multiple loops can be run in the same area on the same frequency.

The transmitter and receiver also have dip-switch adjustable timers. They're set to the same interval. If the receiver doesn't

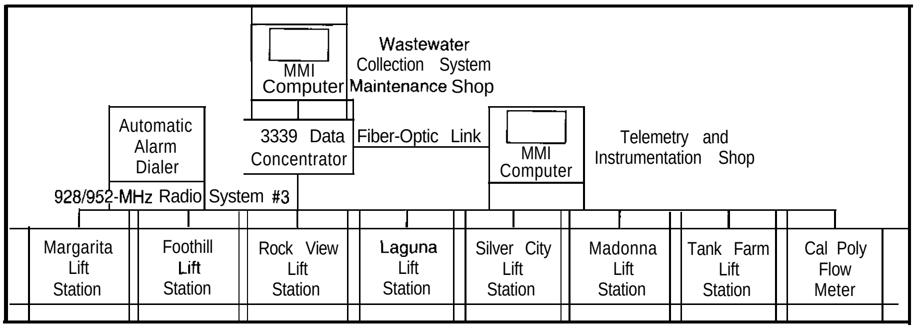


Figure 3: *The* wastewater-collection system uses ***DCS mainly as local controllers with remote monitoring and alarming. In an emergency, the network can manually control lift stations, which have battery backup for wetwell-level monitoring during a power failure.***

receive a valid signal before timeout, a relay closure signalsthat the current output may not be correct and sounds an alarm.

No radio path is perfect, so I set the receiver's alarm time to a multiple of the transmit time to allow for interference. The relay contacts signal the DPC that the current tank level may not be correct. They then send the alarm up the network to the MMI computer to inform operators that there's a problem at the tank.

The older Dataflows were strictly one-way units. The newer model has radio transceivers at both ends and can be used as a simple remote-terminal unit, sending analog and discrete data both ways.

One difficulty with DCS or SCADA systems containing radio data paths is the long lead time and expense in getting a radio license, including the application and frequency coordination.

Available radio spectrum is increasinglycongested, and finding channels that minimize interference to you and other users is harder. A frequency coordinator attempts to do this for new applicants.

SPREAD-SPECTRUM RADIOS

One alternative is license-free radios, which typically use spread-spectrum technology. Rather than a single, distinct narrow-band radio channel, spread spectrum sends the signal over a wide band. It interferes less with other users and is less susceptible to interference.

In one spread-spectrum technique-frequency hopping-the synthesized transceivers simultaneously and rapidly change frequencies in a preprogrammed pattern. Frequency-hopping radios are suited more for direct data input than audio.

In a couple of systems, we use MDS spread-spectrum 902–928-MHz 9310s. These 9310s use the same PC board as the 2310s but different firmware and a frequency-hopping controller in the modem slot. They hop from frequency to frequency at a rate of four per second.

A typical 9310 system has one polling master radio and several remote radios. They are all programmed to use the same channel or block of 64 frequencies, hop pattern, and system address so they can communicate with each other.

When radios on a 9310 system are powered up, the remotes wait for the master's hop pattern to synchronize. Initial sync can take up to 16 s (64 frequencies divided by the 4-frequencies-per-second hop rate).

Once the remote locks on to the master's signal, it speeds up to the master's hop rate. If synchronization is lost, the remote slows its hopping to resync.

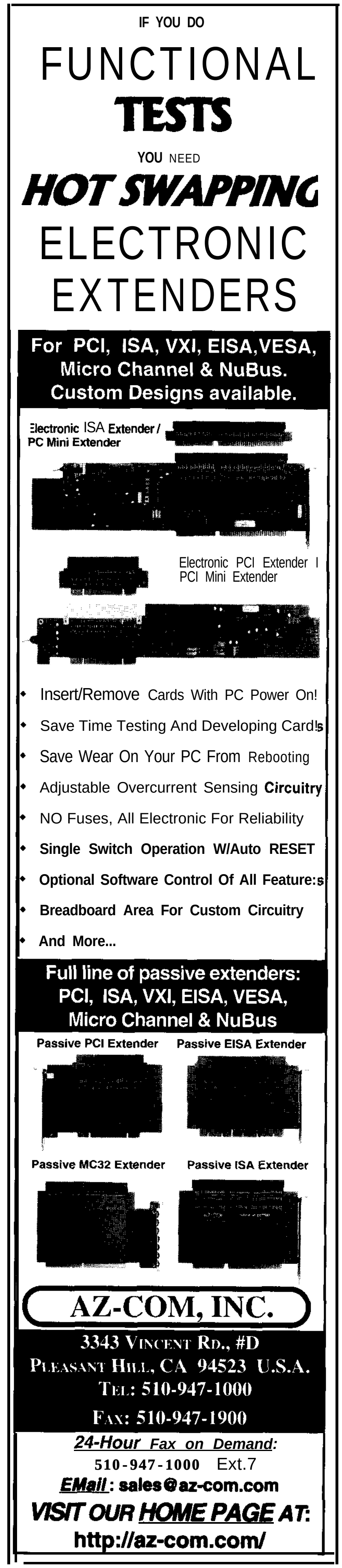
The 9310s internally error check data packets and retransmit garbled ones to ensure clean data is sent between devices. Internal and PC-based diagnostics check link integrity between the master and remote radios to measure the usefulness of individual frequencies.

Like the 2310s, each 9310 has its own four-digit loopbackcode for individual testing. If interference renders one or more frequencies in a given channel unusable, they can be masked out of the hop pattern.

The mask is sent from the master to each remote on command. So, the whole network is updated at once, and there's no need to visit each remote site.

Several frequency-hopping 9310-based systems can be used in a given area if you pay careful attention to channel (frequencies) selection, the hop pattern, and the system address.

One problem with our first 9310-based system was that users hooked the units up using the default channel, hop pattern, and address. If a couple of users in one area are unaware of each other, the 9310 remotes



#203

lock on to the wrong master. Not only won't it work, but according to the carrier-detect (CD) and receive-data (RD) indicators on the radios, it *appears* to work fine. The only way to find out the root cause is to shut off your master radio.

If the remote 93 10 still shows CD and the RD light flashes, your remote has locked on to someone else's master. To avoid this, we routinely change all parameters—not just to the next one in numerical or alphabetical order, but to randomly chosen ones-for each 93 1 O-based system.

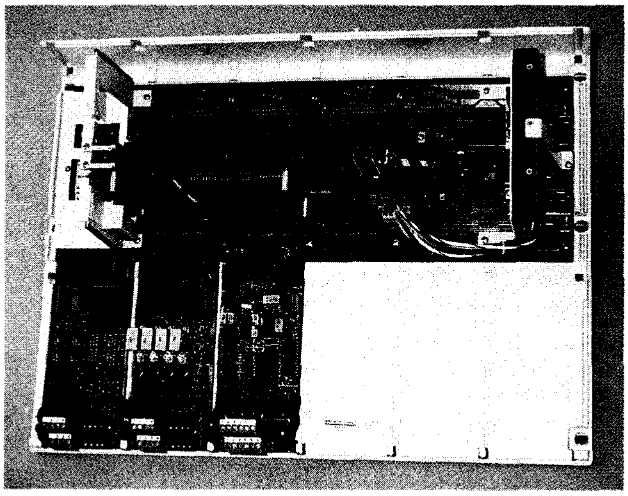


Photo 2: Without a cover, the 3330 shows a bockplane, processor, memory card [the long row of LEDs), and one serial communications card (four LEDs) plugged into it. At the bottom, from left to right, a four-input 120-VAC discrete input card, a four-output relay card, and a four-input analog card fill three of the six slots on this chassis.

REPORT BY EXCEPTION

The data network path for most DCSs is built around a topology that can be traced directly back to copper-wire communications. The Bristol Babcock network is no exception. Its master-slave polling requires a dedicated data path for master-slave communications on each network level.

This network isn't too bad in a plant environment using wire, but it gets unwieldy using radio. The distributed-network design model shares communications among small controller groupings to minimize the effect of any one failure. It clashes with the realities of coordinating and licensing many radio frequencies.

One new system abandons the polling paradigm. Rather than a master station regularly polling for data, they report by exception, sending data only when it changes by a predetermined amount. Data can also be sent on a timed basis so other units on the same data path know a given unit is still active.

The inevitabledata collisions are handled intelligently. So two units don't continuously collide, they wait for acknowledgment and randomize the resend time. Short packets also help.

Data can be addressed to a given recipient or tagged with its unique ID and broadcast for others. Thus, many units can share a common radio frequency. Also, any unit can be a master, asking others for data, and/or a slave, supplying the data.

STORE AND FORWARD

A report-by-exception network adapts to changing network conditions and topology. it's well-suited for difficult locations like mountainous regions or new sites.

Programmed units act as store-and-forward repeaters, passing data between units not in direct contact with each other. A regular radio repeater simultaneously retransmits what it hears on one radio frequency on another frequency.

When reception ends, a store-and-forward repeater records and retransmits what it hears on the same frequency. By addressing messages, tagging them with the address of each intervening unit, and keeping an internal list of locally "heard" units, an adaptive store-and-forward network dynamically minimizes data-path length, bypasses dead or nonresponsive units when possible, and keeps the network running.

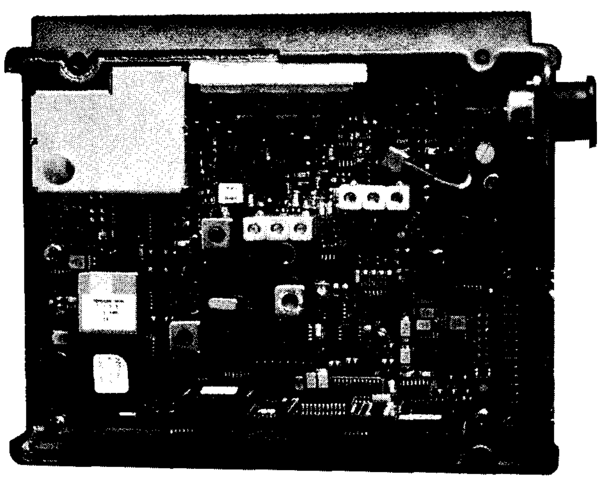


photo 3: looking at the MDS23 10 928-/952-MHz without its cover, in the upper right corner is the 68k-series processor. The two edge-mounted cards on the left are the 9600-bps modem and DTMF remote diagnostics cards.

DIRECT TO DIGITAL

One more development that really puts the distribution in distributed control is direct-to-digital or network-ready sensors.

Digitizing the standard analog transmitter for pressure, distance, or other parameters puts the sensor and network controller in the same package. A remote sensing device added to a communications path without an intervening controller cuts the cost of remote sensing and increases the system designer's options. EPC

Chris *Arndt, KD6DSI, has been the Telemetry and Instrumentation Technician for* San *Luis Obispo's* Utilities Department for 7 years. He designed, programmed, *and* installed **most of** the *city's* DCS systems and is a Grade 5 *State* Certified *Wastewater Plant Operator and Grade 3 Water Plant Operator. You may reach Chris at carndt@slonet.org*.

SOURCES

Network 3000 controllers, ACCOL
Bristol Babcock
1100 Buckingham St.
Watertown, CT 06795
(860) 945-2200
Fax: (860) 94522 13

MDS23 10, **MDS93** 10 radio modems
Microwave Data Systems
175 Science Pkwy.
Rochester, NY 14620
(716) 242-9600
Fax: (716) 242-9620

Dataflow
Ritron
505 W. Carmel Dr.
Cannel, IN 46032
(3 17) 846-l 201
Fox: (3 17) 846-4978

Report-by-exception and store-and-forward technology
Communication Solutions
1261 Talmadge Rd.
Santa Maria, CA 93455
(805) 937-5987
Fax: (805) 937-2919

IRS

416 Very Useful
417 Moderately Useful
4 18 Not Useful

Paul Beam

A Lever and a Place to Stand

Controlling the World with the Internet

Archimedes wanted to move the world, Atlas held *it* up, *and* you-what *do* you *want to* do? *If it's* to have your computers remotely access *all* I/O ports on a server, here's how. Paul shows how *WinSock* brings global access.

One of the first computer books I purchased was TRS80 *as a Controller* by Jerry O'Dell. Years later, it's still on my bookshelf. I always liked the idea of hooking cool stuff to a computer and controlling the world. I guess that's one of the reasons I read INK.

Mind you, the concept of controlling the world is about as far fetched as Archimedes' dream of moving the world with a lever. His dream is fundamentally flawed because it's impossible to have a lever and a place to stand large enough to move the world. Controlling the world with a computer has the similarly imposing problem that everything must be connected to it.

Bringing the concept a bit more into reality, I've spent the past several years connecting computers and other equipment on a small college campus. The advantages of networking are well-known in most organizations these days, but from my perspective, being able to access a remote computer is the most important and exciting aspect.

To be somewhere electronically instead of physically is more than convenient. It opens up all kinds of possibilities.

I've often considered using an office network like the Ethernet for control instead of a proprietary wiring scheme. I'm sure thatcontrol-oriented networks offer specific advantages that make sense for particular purposes.

In applications where tight timing is critical, a specialized architecture is just about the only option. But, for many applications (e.g., building control), I think it's better to use a preexisting computer network.

With the recent boom of the Internet, we no longer have to be content with a locally connected network. The next office and the next continent are about the same distance away, electronically speaking.

Suddenly, if we consider a computer network to be a control infrastructure, controlling the world seems plausible. We have the "place to stand," as Archimedes might put it.

All we need are the simple machines to use it. I think those simple machines could be called TCP/IP.

OLD PROTOCOL LEARNS NEW TRICKS

Briefly, the Internet speaks a collection of protocols known as TCP/IP. Any text about the Internet goes into a lot of in-depth discussion about its military origins and the seven-layer OSI model.

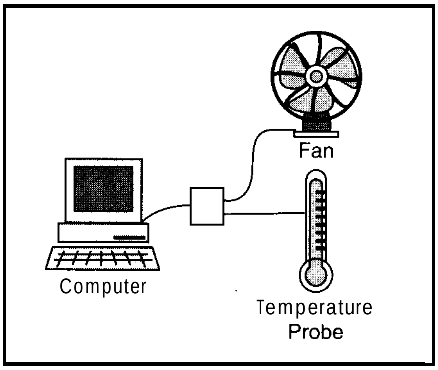


Figure 1: *A single computer attaches to a temperature probe and a fan.*

Since, in my opinion, this history helps little with implementation, I'll skip it. For the initiated, find any other publication on TCP/IP-it will do a better job than I could in detailing its development.

Suffice it to say that, even though TCP/IP is rather old (ancient by computer terms), it has found new life with the commercialization of the Internet.

With TCP/IP, the medium doesn't matter. You might have a dial-up modem connection, Ethernet, Token Ring, FDDI, ATM, or anything else the future holds.

TCP/IP is also implemented on a great number of hardware platforms. It used to be that only UNIX machines spoke TCP/IP, but now it's common on all the Microsoft Windows operating systems as well as many embedded real-time operating systems. It's truly an open system where the transport medium and the OS matter very little.

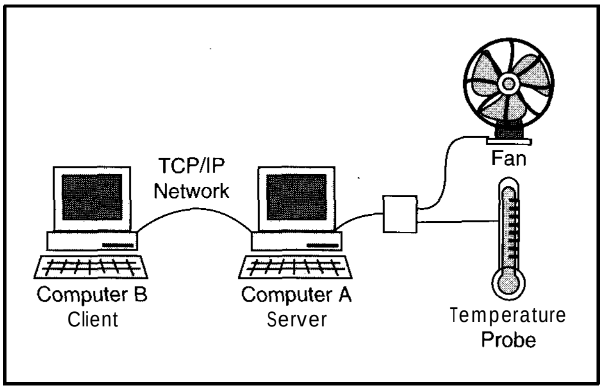


Figure 2: To the ***simple computer in*** Figure **1, I** added a second computer and ***TCP/IP*** network.

Most software implementations, including Microsoft's, are based on the Berkeley sockets model. In this model, a communication channel is created between two machines, and its end points are known as sockets.

One machine is called the server and the other the client. Which machine is the client or server depends on the application. Often, it's valid to call the machine initiating the communication the client, while the machine waiting for a connection acts as the server.

Communication across this channel can either be reliable two-way (TCP) or unreliable connectionless (UDP). The term "reliable" means that protections guaranteeing accurate delivery are built into the protocol. UDP is unreliable in this sense because it lacks protection mechanisms.

Listing 1: Here's some pseudocode of the simple temperature-control system shown in Figure **1. *All*** the system peripherals are attached ***to*** local ***I/O*** ports.

```
/* Is temperature > high threshold? If so, turn on fan */
If (INP(Temperature_Port) > HighTemp)
  OUT(Fanport, FanOn);

/* Is temperature < low threshold? If so, turn off fan */
If (INP(Temperature_Port) < Lowtemp)
  Out(Fanport, FanOff);
```

There's significantly more overhead in a TCP connection, but it's warranted for essential information across a complex network. On a small, lightly loaded, robust network, a UDP packet will reach its destination. But, in a large complicated network like the Internet, a UDP packet may be lost along the way.

Any sort of data can be transmitted through a socket. There's no reason why control data cannot be transmitted like any text document.

Some network devices are controlled by protocols like the Simple Network Management Protocol (SNMP), but **I've** found these to be far from "simple" and much more complex than necessary for simple control tasks.

I envision a truly simple protocol based on /he input and output model that's the bread and butter for most microcontroller programmers. In a control application, it's pretty common to read and write data to or from an I/O port.

Figure 1 shows a typical hardware configuration for a simplistic temperature-control system. A computer monitors the temperature, and if it's too warm, it turns on a fan. When the temperature drops, the fan turns off. The code for this elementary system might resemble listing 1.

To make things more interesting and useful, I'll expand the system as shown in Figure 2. All I'm adding is a second computer and a TCP/IP network.

To make things work remotely, I installed software on computer A to make it a server and software on B to make it a client. With the proper design, devices attached to computer A are available to computer B.

The code, which executes on the client, is essentially the same as the original and is shown in Listing 2. The only changes are replacing `OUT()` functions with `NET_OUT()` and `INP()` functions with `NET_INP()`.

These new functions have the additional parameter of a remote machine name, as well as the I/O port. This feature effectively allows all I/O ports on the server to be accessed by the client, enabling the client to control devices attached to the server.

This model is easily expanded to more clients and servers. The TCP/IP network could be replaced with the Internet. Pretty soon, there's a global pool of accessible I/O ports, putting world-wide control almost within reach.

PLUG IN TO SOCKETS

Sockets programming is mostly cookbook-style programming. All server applications have an essentially identical block of code. The code in Listing 3 is the minimum necessary to create a server application.

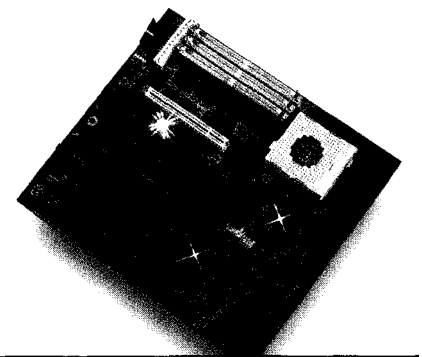
A commercial implementation provides more robust error detection. Where servers differ is in the area of packet interpretation.

listing 2: The client code from listing **1** is adapted to the network system ***shown in Figure 2.*** The local *INP* and ***OUT*** instructions are replaced ***with network*** *NET_INP* and *NET_OUT* ***to*** provide remote access.

```
/* check temp given by RemoteMachine. If too high, turn on fan */
If (NET_INP(RemoteMachine, Temperature_Port)   >   HighTemp)
  NET_OUT(RemoteMachine, FanPort, FanOn);

/* check temp reported by RemoteMachine and turn off fan */
If (NET_INP(RemoteMachine, Temperature_Port) < LowTemp)
  NET_OUT(RemoteMachine, FanPort, FanOff);
```

• 7 ISA slots
• 486 support



The **G486VPC** motherboard combines quality and affordability with an industrial design to meet your needs. Seven full length ISA slots ensure expandability for the cards that you use.

Our engineering staff will gladly discuss custom motherboard designs. FCC & UL certified systems are also available.

- Six full length 16-bit ISA, One shared 8-bit ISA/PCI slot
- Intel, AMD & SGS-Thomson 486 CPU support
- VIA chipset
- Up to 64MB RAM, 256KB cache

DFI®

#205

listing 3: The code necessary to create a server *is essentially* unchanged from *BSD sockets to* ***WinSock. WinSock*** requires some additional initialization and *cleanup not* shown.

```
char myname[32];          /* holder for machine ID */
char buffer[150];         /* send/receive buffer, make good size */
SOCKET s, connected_socket;                /* socket descriptors */
int portnum 4567;         /* TCP/IP port to listen on */
struct hostent *hp;       /* host entry structure */
int ReceivedBytes         /* received packet size */
struct sockaddr_in sockaddr;               /* Inet-style socket addr */
memset(&sock_addr, 0, sizeof(sock_addr)); /* clear addr */
gethostname(myname, MAXHOSTNAME);          /* get own name */
hp gethostbyname(myname, MAXHOSTNAME); /* get host addr info */
sock_addr.sin_family AG_INET;              /* set addr type to ARPA */
sock_addr.sin+port =3D htons(portnum);   /* set port to listen on */
s socket(AF_INET, SOCK_STREAM,0);          /* create TCP socket */
bind (s, &sock_addr, sizeof sock_addr); /* bind socket/host addr */
listen(s,3);              /* listen for conn, queue up to 3 */
connected_socket accept(s, NULL,NULL);
/* wait for conn, return socket descriptor for conn */

Received Bytes recv (connected_socket, buffer, sizeof buffer,0);
/* receive bytes and put them in buffer */
  [Process received packet herel

  [Create Packet to send here]
send (connected_socket, buffer, sizeof buffer,0);
/* send buffer to connected machine */

closesocket(connected_socket);             /* close conn with remote */
```

The application-level protocol is what distinguishes one server from another.

In every application, both the server and client must create a socket through which to communicate. In UNIX, a socket and a file are the same, but in DOS, a file handle and a socket descriptor are two entirely different animals.

Much of the work in creating a socket involves setting up the address information. Every machine on a TCP/IP network has a unique name and IP address.

Although we usually refer to machines by name, computers refer to each other by number. Fortunately, all we have to do is call the correct sequence of functions to translate names into numbers and then translate the numbers into the proper byte order.

After the socket is created, the server will `bind()` it. The server can then `listen()` and accept () incoming connections. With a connected socket, it's rather straightforward to implement and to send () and `recv()` dialog with the client.

I kept things simple and defined a protocol consisting of three different packets. The packets transmitted by the client are shown in Figure 3.

The OUT packet is four bytes consisting of the letter "0" followed by a two-byte address and single data byte. The I NP packet is only three bytes-an "I" followed by a two-byte address.

For simplicity, I only implemented eight-bit instructions, so the only packet the server transmits is a single return byte to the I N P instruction. One-, three-, and four-byte packets may not make efficient use of bandwidth, but it's all the complexity this application requires.

I also chose TCP instead of UDP, which takes even more bandwidth. In most applications, it's rather important that the packet arrive as intended. And, not being a real enthusiastic programmer, I'd rather the protocol do the error checking for me.

One last (and potentially confusing) element of the server is that a server is bound to a particular TCP port, which has to be distinguished from an I/O port. I wish there were two terms to distinguish between them.

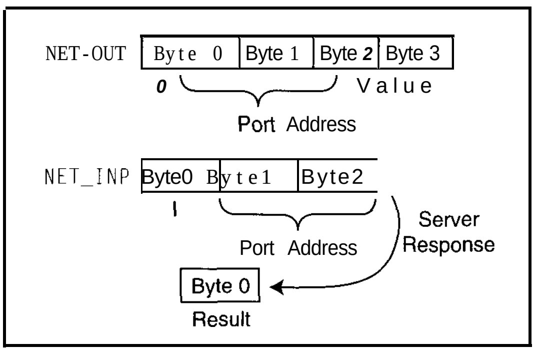


Figure 3: This figure shows the ***structure of the data packets, which are*** *basically just boxes indicating* ***the*** *byte order.*

I chose port 4567 because it's easy to type and, as far as I know, not reserved for any other application. Thus, a client must establish a TCP connection to the server on port 4567 to gain access to the I/O ports on that server.

The client is also like a cookbook. The code in Listing 4 is all that's required for a basic TCP client.

Many of the structures and functions are identical to those for the server. A socket still has to be created, but it must be based on the server's address and TCP port instead of the local machine's, Instead of a `listen()`, the client uses `connect()` to call the server and establish connection.

The structure of the I/O client is rather different. I'm intending the client to be general purpose and easy to use. It makes sense that the client should be a couple of functions that can be compiled into applications for accessing remote devices.

Thus, instead of a stand-alone program, it should be a library of routines that other programs can use. At this time, the only functions are `net_inp()` and `net_out()` for network input and output, respectively.

SERVING UP I/O PORTS

Implementing the I/O client and server requires the choice of an operating system. Given that the server must access I/O ports directly, a protected OS is ruled out unless a suitable virtual device driver (VxD) is available.

Since I didn't want that additional complexity, 16-bit Windows looked like the best approach. Windows may not be the best choice for an embedded system, but since the server can run as a background task, it makes any office PC a potential candidate. Besides, an old Windows-capable machine costs far less than it used to, and developmenttools are readily available.

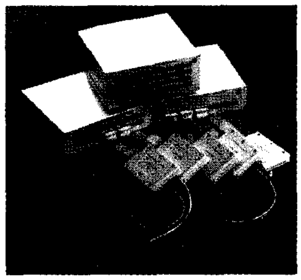
Microsoft defines a socket TCP/IP specification called WinSock. The WinSock specification is available from Microsoft's ftp site. (While there, pick up winsock.h and winsock.lib, too.)

WinSock is included with Windows 95 and Windows NT. There's a drop-in protocol for Workgroups and several third-party implementations for Windows 3.1.

The WinSock specification differs slightly from the original BSD sockets, but it's similar enough that programs can be easily ported to or from a Berkeley socket system.

Serious Test Instruments

500MHz Logic Analyzers

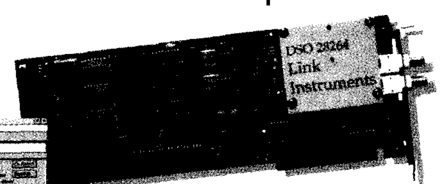


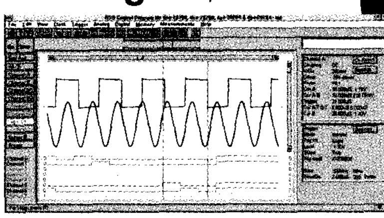
Starting at $1350

Get the **speed you** need with our instruments. Our 500 MHz PC based logic analyzers with up to 160 channels, 512K of memory per channel, and optional digital pattern generation.

200MS/s Digital Oscilloscope & logic Analyzer

Starting at $1799

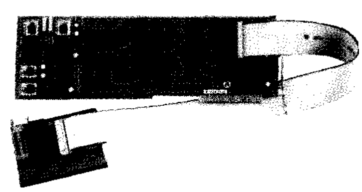




Our 200 MS/s DSO gives you 2 or 4 scope channels with long 128K memory buffers, 8 or 16 channels of logic analysis, and FFT spectrum analysis, all integrated into one card. Easy hardware setup and straight forward DOS and Windows software get you started right away.

Model 3100 Device Programmer

$475 - Our device programmer 3100 programs PLDs, PAL, GAL, EPROM, Flash, serial memories, and microcontrollers.



Link Instruments
(201) 808-8990
369 Passaic Ave . Suite 100 . Fairfield, NJ 07004 . Fax (201) 808-8786
Web: www.LinkInstruments.com . Email: Sales@LinkInstruments.com

#206

EPC

Even though Microsoft doesn't recommend it, the code in Listing 3 and Listing 4 can be placed directly into a QuickWin application.

The reason Microsoft does not recommend it is because functions such as call() and accept() are blocking calls by default. A blocking call sits and waits until the function completes before it returns control back to the calling program. In a cooperative multitasking environment, this isn't recommended, but it's fairly common.

Windows requires a WinSock application to call WSAStartup() before using any WinSock functions and WSACleanup() before the application exits. The little extra housekeeping is shown in Listing 5.

I created the server with Microsoft Visual C++ V.I .O. Granted, that's a pretty dated version, but it was collecting dust on my shelf. The code should work with other compilers that have an environment like QuickWin.

QuickWin enables DOS-like programs to be developed for Windows. The old familiar printf functions are there, among others, without the overhead required by Windows. I've found that compiled as a 16-bit application, direct access to I/O ports is still available under Windows 95.

The client is fashioned after the code of Listing 4. The Windows startup and cleanup code is part of each function, so your native code doesn't need to know anything about networks. It could be compiled as a DLL and used with your favorite language without great difficulty.

Visual Basic custom controls with these routines would make an impressive display without much work. A couple sample applications you can use to amuse your friends are also available.

Listing 4: Here's all the code necessary **to** create a sockets **client.**

```
SOCKET  s;                             /* socket descriptor */
struct sockaddr_in sock_addr;          /* ARPA-style socket addr */
struct hostent *hp;                    /* host entry structure /*
int tcpport 4567:                      /* TCP port to connect to */
char server[32]'Server';               /* server name to connect to */
char buffer[150];                      /* packet storage buffer */
int ReceivedBytes;                     /*# bytes received from server */
memset(&sock_addr, 0, sizeof(sock_addr)); /* clear addr */
hp gethostbyname(server);              /* get addr of server */
memcpy(&sock_addr, hp->h_addr, hp->length); /* copy addr */
sock_addr.sin_family AF_INET;          /* set Internet type */
sock_addr.sin_port tcpport:            /* port to connect to */
s socket(AF_INET, SOCK-STREAM, 0);       /* create socket */
connect(s, &sock_addr, sizeof sock_addr); /* connect to server */

send (s, buffer, sizeof buffer, 0);   /* send buffer[] to server */
...
ReceivedBytes recv(s, buffer, sizeof buffer,0);
/* receive buffer[] from server */
```

INTO THE FUTURE

Using this client and server should add new life to old control projects. It's not necessary to totally redesign hardware for remote operation. Systems built to monitor and control can now do it remotely. And with the Internet, remote can be very remote indeed.

These programs are not, however, suitable for common use in an Internet environment because there's no security whatsoever. Any computer on the Internet can access any I/O port on any machine this server code is executed on.

Think about someone messing with your hard-disk controller or the frequency settings of your video, and you see the risk. This application is very powerful but potentially very dangerous. Some rudimentary security based on access-control lists would be a great improvement if **you** implemented this system in a public network.

This system is also not suitable for real-time applications because propagation delays of packets across the Internet can't be reliably determined. Usually, packets are delivered within a few hundred milliseconds, but this can stretch into tens of seconds if it's really busy.

New techniques being developed for the real-time transmission of audio and video might provide a more deterministic environment. Operation on a local network could be suitable for a great many applications.

One logical extension of this application would be to replace the client with a Web browser with suitable client programming on a Web server. Another possibility is to remotely load and execute a program on the server instead of simple I/O operations. Once the TCP connection is established, interpreting the information packet is the only difficulty.

I hope you've seen some of the possibilities of using networks for real-world control. If Archimedes were alive today, you might hear him say, "Give me TCP/IP and the Internet, and I'll move the world." EPC

After Paul Beam earned his MSEE, he began work ***as a*** computer systems engineer *at Johnson* Bible College in Knoxville, TN. You may reach **him at** *paul@jbc.edu*.

listing 5: ***This*** code is necessary for ***WinSock*** setup and cleanup.

```
/* Startup */
long RetCode 0;
WSADATA wsaData;

/* check for WinSock 1.1, 0 means WinSock found Cleanup */
RetCode WSAStartup(0x0101, &wsaData);

/* Windows Sockets Cleanup */
WSACleanup();
```

SOFTWARE

The complete source code for this article is available from the Circuit Cellar BBS, the Circuit Cellar Web site, or Software on Disk for this issue. Please see the end of ConnecTime for complete downloading and ordering information.

REFERENCES

C. Hunt, *TCP/IP Network Administration*, O'Reilly and Associates, Sebastopol, CA, 1992.
Microsoft Corp., "Windows Sockets An Open interface for Network Programming under Microsoft Windows V. 1.1," ftp://ftp.microsoft.com/bussys/winsock/spec1 1, January, 20, 1993.

SOURCES

WinSock
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
(206) 882-8080
Fax: (206) 936-7329

IRS

419 Very Useful
420 Moderately Useful
421 Not Useful

Derek Seward and
Stephen Quayle

"wanna be famous?" WINNER PC/104 Design Contest

LUCIE

Lancaster University Computerized Intelligent Excavator

Using three PC/ 7 04 embedded computers, *LUCIE* makes the intelligent decisions of a human operator. Route planning, priorities, navigation, depth alterations, safety *considerations—all* are part of how *LUCIE* digs a trench.

In 1987, the Lancaster University Engineering Department combined the three disciplines of civil, mechanical, and electrical engineering in to one "mechatronic" project-the Lancaster University Computerized Intelligent Excavator. LUCIE, as we've come to call it, is shown in Photo 1.

Over the next few years, we developed a JCB801 minitrackedexcavatorthatcould dig a good-quality trench in highlyvariable ground. Theexcavatorwas positioned manually before excavation began. Due to safety concerns, we didn't automate the tracks.

The system basics consisted of a parallel set of electrohydraulic valves connected to a large "controller box" filled with various DAC, ADC, signal-conditioning, and safety **cards.** Software was written in Forth, and arm-position feedback was achieved via digital optoencoders.

In 1994, we updated the excavator's hardware. The entire hydraulic system was stripped and replaced with a set **of** Danfoss electrohydraulic PVG32 proportional control valves.

Photo 1: LUCIE, the Lancaster University Computerized ***Intelligent*** Excavator, is undergoing ***tests*** to check out ifs autonomous trench-digging capabilities.

the controller box was replaced with a Danfoss EHDC controller. Unfortunately, this controller couldn't cope with the high computational demands. So, we replaced it with three PC/l 04 units.

Work started on navigation and automation of the vehicle tracks. Two Trimble 7400MSi satellite receivers in differential mode achieve positional precision at the centimeter level.

Safety concerns led to the development of a research program-safe System Architectures for Large Mobile Robots, funded by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under the DTI/EPSRC Safety Critical Systems Programme.

The brain of the excavator relies on the three embedded PC/l 04 computers. Each one is responsible for a separate task. Figure 1 shows how these processor stacks connect. We'll discuss each stack in turn.

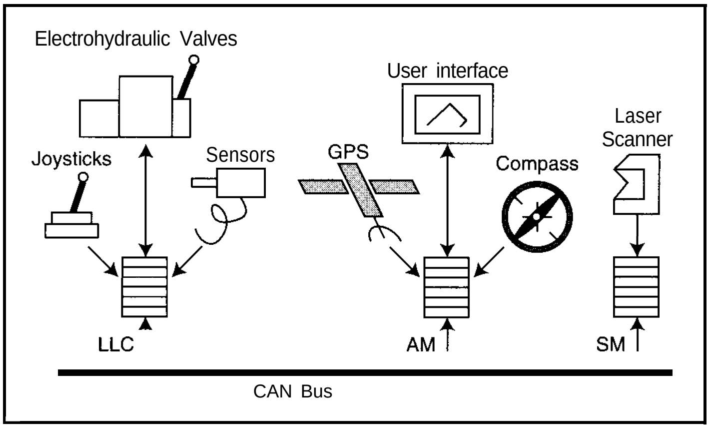


Figure I: Three **PC/104** stacks control the **excavator.** Communication between the ***stacks is*** achieved via CAN bus. The eventual aim ***is to have*** all sensors and actuators ***acting os nodes directly from the*** CAN bus.

LOW-LEVEL CONTROLLER

The Low-Level Controller (LLC) diagrammed in Figure 2 is the largest of the PC/l 04 stacks. It has a 80-MHz Tiny '486DX processor card from Advanced Micro Peripherals with 4-MB DRAM and 2-MB flash disk.

The DM5400 ADC I/O card from Real Time Devices has 16 I/O channels, a 12-bit ADC, 16-TTL/CMOS 8255 DIO lines, three 16-bit counters, two DAC channels, and diagnostics software. Real Time Devices' DM5604 DAC I/O card has eight 12-bit DAC channels, 24-TTL/CMOS 8255 DIO lines, and diagnostics software.

The MSMCAN CAN bus card from Digital-Logic includes an Intel 82527 controller and achieves 125 or 500 kbps. The Power Distribution II from parvus has 7.5-17 VDC input and 20-W combined output.

The LLC's main responsibility is controlling the hydraulic valves. There are three different modes of operation.

In the direct mode, joystick inputs via the ADC card are mapped directly out to the valves via the DAC card. This gives control of the excavator in the conventional manner (i.e., 1° of freedom of the joystick corresponds to 1 valve operation). If necessary, the PC/l 04 stack can be removed and the joysticks connected directly to the valve drivers.

The soft-stop mode is like direct mode, except the arm position is monitored using simple rotary potentiometers fixed across the arm joints. These potentiometers connect to the ADC card, which reads each channel 20 times before averaging the results. (This eliminates noise and voltage spikes, giving accurate computation of the arm's position.)

If the processor sees the arm reaching its mechanical limit, it proportionally closes the valve producing a soft-stop effect. Various joystick gain settings are also possible, making intricate maneuvering of the arm very simple.

XY mode is the most complex of the three modes. A series of forward and reverse kinematic calculations make it possible to move the excavators bucket in a straight line.

This task is done by setting a series ofvirtual (target) points for the excavator. The difference between the current and target positions (i.e., positional error) forms the basis for the drive commands to each valve.

If the arm is deflected off course (e.g., by an underground obstruction), the positional error changes, modifying the drive commands. Thus, the system becomes self-rectifying.

Moving one joystick excavates a flat-bottomed trench to within 25 mm-a feat normally requiring much time and a highly skilled operator. XY mode is used for the automatic excavation routines.

ACTIVITIES MANAGER

In effect, the Activities Manager (AM) replaces the human operator. It plans activity (e.g., which trench to dig first) and navigation and is diagrammed in Figure 3.

The AM's processor card is a Tiny '486 Slc, running at 33 MHz with 2-MB DRAM. The VGA1 04 is a 5SD display card from Advanced Micro Peripherals. It has5 12-KB display memory and a 1 -MB flash disk. LIke the LLC, the AM uses the MSMCAN CAN bus card.

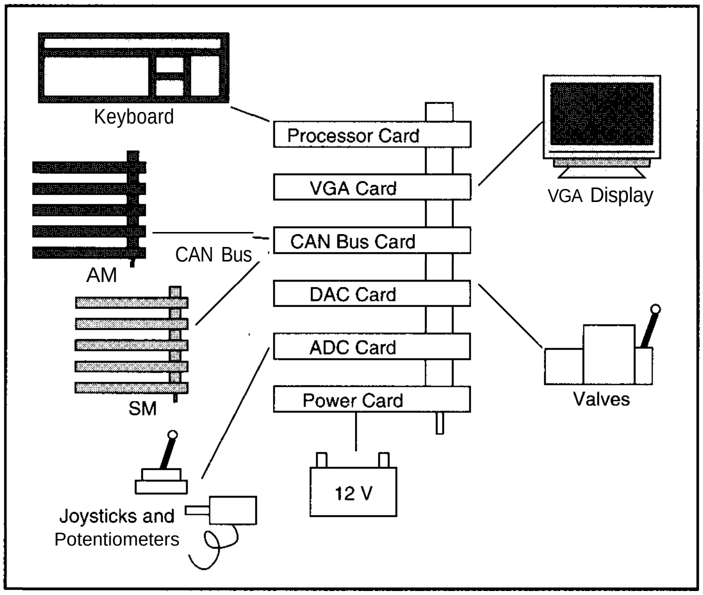


Figure 2: Six **PC/104** cards connect together to form the Low-Level Controller.

The AM has a Power Distribution with 7.5-l 7-VDC input and 20-W combined output. It also includes an M-Systems 2-MB flash disk.

For positioning, the AM uses a Trimble Navigation 7400msi GPS. It has centimeter accuracy, real-time positioning, 5-Hz updates, and automatic OTF (on-the-fly) initialization while moving. It also includes four serial I/O ports and NMEA-0 183 message outputs.

The Satellite-2ASx Radio Data Modem from XL Systems has a frequency range of 405–470 MHz, 16 software-selectable channels, an RS-232 interface, and 1200-9600 bps.

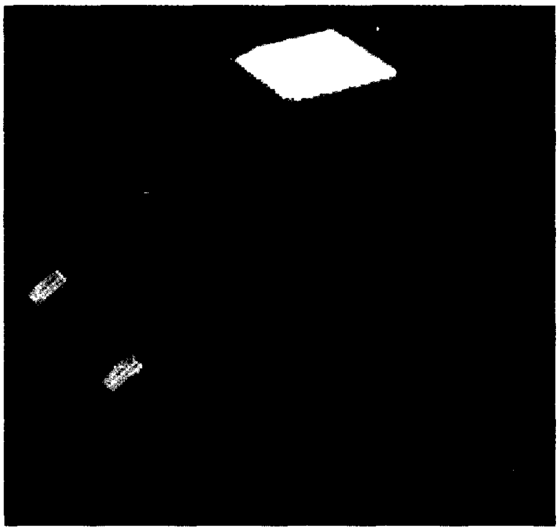


Photo 2: We used the Star Track vehicle to develop navigation principles using GPS.

Automatic dig routines developed to mimic joystickcommands are passed to the LLC operating in *XY* mode. This returns data on current arm position and status.

By monitoring the rate of the arm relative to the given commands, the AM deduces if the ground is hard or soft and changes the cut depth accordingly. This feature makes the digging process more efficient and enables successful removal of obstructions.

The AM program structure is based on an artificial-intelligence technique known as a production system. The system is divided into three parts.

The production-rule memory contains a set of if-then rules. A typical rule for LUCIE is:

```
IF (penetration of bucket > 300 mm)
  THEN (rotate bucket and go to tip)
```

The working memory contains the data necessary to check the conditions in the left-hand side of the rules. All the latest sensory data is stored for other parts of the program. In the future, the working memory may be replaced to a large extent by having data available directly from the CAN bus.

The third part, the inference engine, cycles through the rules, checks their conditions, and fires or sanctions them if necessary. Using this system lets rules be added or changed without significantly affecting the rest of the code.

NAVIGATION

The AM connects to a differential global positioning system (GPS) and compass. We developed the navigation system and real-time vehicle control using a small tracked vehicle known as Star Track (see Photo 2).

Star Track uses a Trimble 7400msi GPS operating in differential mode. It provides positional information accurate to 25 mm and at a 5-Hz update rate. With the GPS and data from an onboard compass, Star Track knows exactly where it is and—just as important-which direction it's facing.

Once the tracks have sensors and movementscan be dead reckoned, LUCIE will be able to travel under bridges and continue to work when the GPS signal is lost.

To navigate around a site, we set up a series of intermediate way points. These consist of a GPS location and a path width (i.e., the distance the vehicle can veer safely from the stored location).

LUCIE calculates tangent points for arcs to intersect the straight line paths, keeping within the given constraints of path width. Therefore, the vehicle moves more fluently, cutting corners in an almost human-like motion as shown in Figure 4.

We plan to add a higher level controller that creates alternative intermediate points should the original path become blocked.

We hope to use a production system similar to the XY bucket control to keep the excavator on its designated path. By doing this, a data table can be used to simplify the navigation program.

We're investigating the postprocessing of CAD drawings to create this table automatically. The draftsman's drawings then go directly to excavation instructions without human intervention.

USING GPS

Most commercial GPS systems use the U.S. Defense Department system of 24 SATNAV satellites (although there is an alternative Russian GLONASS system). These orbit at about 20,000 km, and at any one time, you can see 5-l 1 satellites.

We'll briefly review the GPS basics here, but see Do-While Jones' series on GPS (INK 77 and 78) for details.

The basis of positioning is the accurate measurement of distance from a terrestrial receiver antenna to several satellites. This distance is determined by how long a satellite signal takes to travel to the receiver.

Knowing the distance from one satellite fixes the position to the surface of a sphere centered on the satellite. Having the distance to two satellites locates the position on the surface of a circle formed by the intersection of two spheres.

The distance to three satellites reduces the position to two points on the circle's circumference. In theory, knowing the distance to four satellites defines the coordinates of a specific point.

However, distance measure ments are subject to numerous errors. Therefore, modern systems usually prefer six satellites and use sophisticated averaging techniques to get the best estimate of the position.

A complication arises because the satellites emit signals in two codes on different frequencies. The Coarse Availability (C/A) code is the signal used by yachtsmen and the cheap hand-held receivers used by climbers.

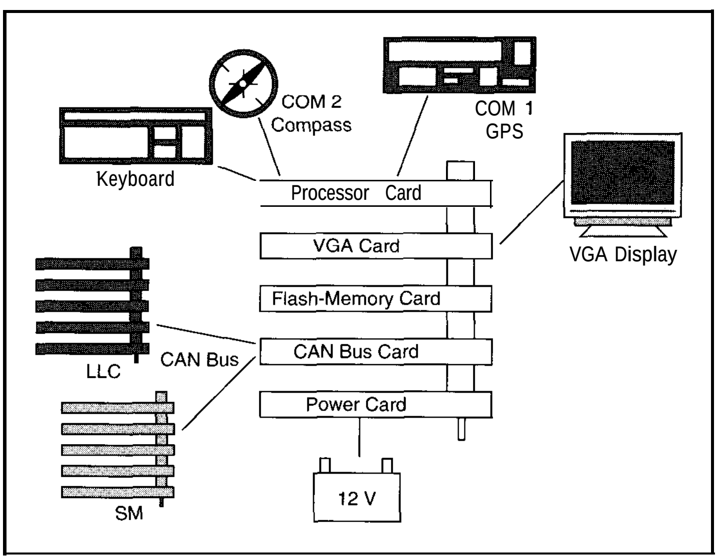


Figure 3: The Activities Manager collects positional information from the compass and GPS to broadcast *on* the CAN bus.

The P-code (Precise code) is subject to selective availability (SA), deliberately introduced errors which can only be decoded by authorized military users. Thus, the signal's accuracy is eroded for ordinary users [1].

In nondifferential mode, the 7400MSi receivers determine location via range measurement. Positional accuracy is dictated by SA and is typically within 100 m. This can be improved if the system has time to settle. Without SA errors, P-code users can achieve an accuracy of -1 m.

Operation in differential mode increases positional accuracy **via** two interconnected receivers. One receiver, known as the base station, is set up at a known survey point. It calculates the difference between its position and that indicated by thesatellites. This difference is transmitted to the rover receiver via radio modem as shown in Figure 5. By differentiating the satellite range measurements, the accuracy with the two receivers becomes -1 m.

The distance between the base station and rover is currently limited to ~ 10 km, which ensures that the two receivers have the same set of satellites and atmospheric errors. One base station can serve several rovers. A strategically placed base station can provide for a whole city.

To achieve centimeter-level accuracy, the rover receiver tracks carrier-phase data and initializes itself relative to the base station. Initialization resolves the integer carrier-phase ambiguities between receivers and satellites.

Before initialization, the 7400MSi receiver provides float **solutions,** differentially corrected relative to the base station and accurate to -1 m. Once initialized, the mode switches from float to integer fixed, and precision drops to the centimeter level.

The 7400MSi receivers are most accurate when operating at an update rate of 1 Hz, so this value is used for final positioning. A 5-Hz update rate is preferable for fast-moving vehicles where the reduced latency better indicates a true position.

The transmission speed of the RTK base-station data to the rovers is important to system accuracy and solution latency. Measurement data, base-station coordinates, and a base-station description message are sent via the data link.

The measurement data requires less than 2000 bits to transmit information for 9 satellites, hence a 2400-bps modem can support an update rate of 1 Hz. It's preferable to use a faster transmission rate (e.g., 9600 bps) as on LUCIE's Satellite 2ASx radio data modem.

The base receiver must be positioned at a known location and have its GPS coordinates set manually. The rover location is computed relative to the base station. Every 10 m of error in the base-station coordinates introduces a 1 ppm in the baseline vector. So, a position error of 50 m with a baseline vector of 10 km produces an error in the rover location of -5 cm.

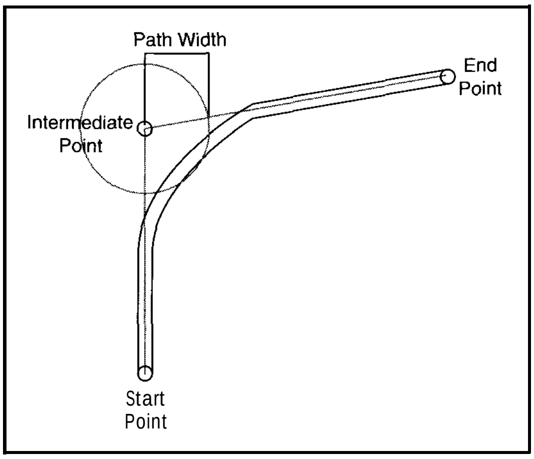


Figure 4: To move to *a* set ***location,*** o series of intermediate ***points is*** used. Stor Track cuts corners, producing o more human-like motion.

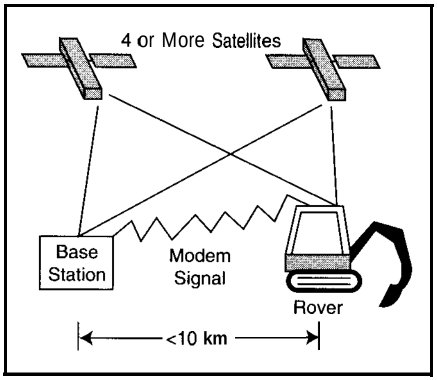
The term "real-time kinematic" refers to the ability to rapidly update the position of moving receivers. Obviously, it's a requirement for control applications. When a receiver first switches on, it takes some time to initialize itself (currently about 2 min.).

If a receiver loses contact with the satellites for any reason, it must reinitialize. The term "on the fly" refers to the receiver's ability to reinitialize without returning to a previously known point [2].

The signal from the rover receiver on the excavator passes to the PC/I 04 stack via an RS-232 serial data link. The 7400MSi data string has many possible options, including choice of ellipsoids, velocity data, magnetic bearings, date and time settings, local datum positions, and so on.

#208

Figure 5: Using a base station and rover achieves differential GPS positioning with centimeter-level accuracies. SA errors are removed by the base station andpassedto the rover via radio modem.



Options currently used are:

- A North-the north component of a vector from the base station to the rover projected onto a plane tangential to the WGS84 ellipsoid
- A East-the east component of the same vector
- A Up-the difference between the ellipsoid height of the tangent plane at the base station and a plane parallel to this passing through the rover point
- GPS quality indicator

SAFETY MANAGER

The Safety Manager (SM) acts as the excavator's conscience, ensuring that the machine remains in a safe stable condition. To detect obstacles, the SM is connected too laser-scanning sensorvia RS-232 (see Figure 6).

The SM uses the Tiny '486 Slc and the MSMCAN CAN bus card. like the AM, the SM uses a Power Distribution I and an M-Systems 2-MB flash disk.

The SM monitors the state of the other two controllers via the CAN bus and is capable of taking the AM offline and providing instructions to the LLC itself.

The CAN bus protocol was specifically designed so the SM has priority over all other systems. If the AM fails to respond to a shut-off command, the SM messages override and get through to the LLC.

If an unsafe state is detected, the SM decides on the best course of action. A situation may arise where a sudden stop is more hazardous. Theexcavator mightwithdraw the arm before stopping, leaving it in a more stable state.

ROTOSCAN SENSOR

The principal sensor available for object detection within the working envelope is a sophisticated scanning laser known as a Leuze Rotoscan RS3. It works by using two losers which scan through 90°, providing a semicircle of coverage as shown in Figure 7.

Objects greater than 7 cm wide are detected up to 25 m. The area is swept at 10 Hz, and the precise position of objects is output in serial form to the computer.

CAN BUS COMMUNICATIONS

CAN bus directs communication between the cards. This communications bus developed by Robert Bosch Co. in the mid '80s was designed primarily for the automobile industry but is gaining acceptance in many fields.

It is defined by ISO 1 1898 and ISO 1 15 19 and uses two signal wires and a power wire. (Systems are currently being developed to pass signals down the power wires). The full CAN bus protocol is very fast, with transfer rates of up to 1 Mbps.

No addresses are given in the message identifier. Instead, the message is labeled with its contents (e.g., RPM, hopper full, etc.). Messages are then only picked up by nodes requiring that information (i.e., message filtering).

The message identifier determines the priority using nondestructive bitwise arbitration. It uses nonreturn-to-zero encoding, so if a signal wire is broken, communication continues.

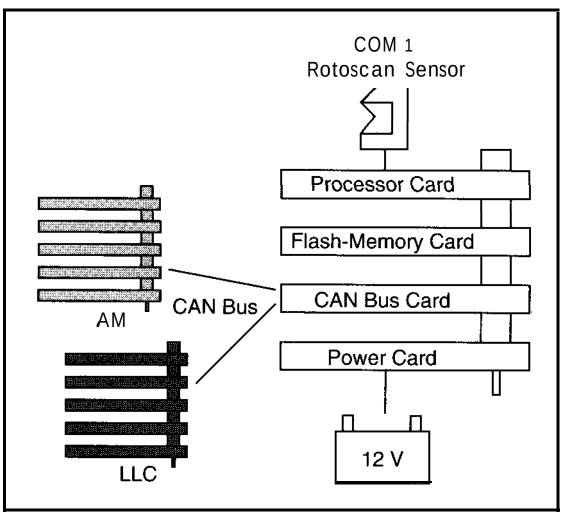


Figure 6: The ***Safety*** Manager uses the Rotoscan to detect unforeseen objects. It can stop the excavator ***by*** sending commands across the CAN bus.

Also, it contains five error-checking mechanisms as well as an error-confinement feature unique to CAN. It detects if a node is faulty, and if so, it is turned offline. The defective node is monitored, and if it becomes OK again, it is turned back on.

It is estimated that a CAN operating at 1 Mbps (with 50% utilization, using an average message length of 80 bits, and operating for 8 h per day for 365 days a year) will contain one undetected error once every 1000 years.

Currently, only interPC communication is via CAN bus, but we intend to gradually move all sensors and actuators onto the bus. CAN microcontrollers can be used as nodes to feed clusters of sensors or actuators, but the technology isn't trivial and there's an extensive learning curve and the need for software CAN analysis tools. Items such as rotary encoders are now coming on the market with embedded CAN bus interfaces.

FUTURE DEVELOPMENTS

Although we've done a lot of work on LUCIE, much remains. The environment of an excavator is different from a standard robot's, so LUCIE's system requirements are much greater.

The possible interaction with human beings and the machine's enormous power make safety o critical issue to be solved before automatic excavation can become reality.

But, there are many advantages to an automatic system over manual-control methods, including an increase in the speed of trench digging and improvements in the dimensional accuracy of the trench.

There's a reduction in the operator skill levels necessary, thereby opening up the backhoe market to less skilled customers. The use of remote control reduces the need for bank persons.

The computer can display any information required about the trench and arm position.

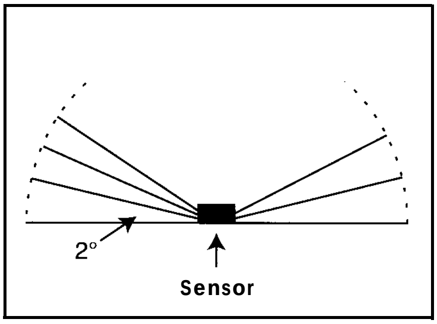


Figure 7: *The* **Leuze** *Rotoscan sensor uses* **two** *scanning laser* beams to detect *obstacles up* **to** 25 *m away at angular increments of* 24

Increasing speed and efficiency may reduce a construction company's lead time, providing them with a competitive edge.

Of course, these machines aren't limited to the work schedules and conditions that people are. And, by reducing human labor and using power more efficiently, running costs are reduced.

As well, autonomous operation is possible in hazardous environments such as chemical accidents or the decommissioning of nuclear power stations.

Automation preserves the vehicle itself by incorporating such features as "soft stops" on ram operations and condition monitoring. It also improves safety via operator warning systems and prevention of movement outside a safety envelope with tilt sensors and metal detectors.

However, our most immediate concerns for LUCIE **involve extending CAN bus to incorporate all sensors. We'll also be linking task planning** to CAD drawings, **as well as doing more research into underground obstruction sensors and SAFE-Sam.** PCQ.EPC

Derek Seward is a lecturer in Engineering *at* **Lancaster** *University.* **He teaches software engineering and intelligent systems and conducts research in construction robotics and** *automation. He has authored over thirty publications and has a particular* **interest** *in the safety of large autonomous robots. You may reach* **Derek at** *d.seward@lancaster.ac.uk.*

Stephen *Quayle graduated from Lancaster University* **with a Masters in** *Mechatronics. After completing the Advanced Course in* **Design, Manufacture, and Management run jointly between Cambridge, Durham,** *and Lancaster Universities, Stephen returned to Lancaster to help develop an autonomous backhoe excavator.*

REFERENCES

[1] Groover, Weiss, Nagel, Odney, *Industrial Robotics,* McGraw-Hill Publishers, 1986.

[2] D.W. Seward, J.R. Ward, J.D. Findlay, and H. Kinniburgh, "The Automation of Piling Rig Positioning using Satellite GPS," 13th Intl. Symp. on Robotics in Construction, Tokyo, June, 1996.

Journal of PC/104 Controlled Systems, http://www.controlled.com/pc104/

Robotics at Lancaster University, http://www.comp.lancs.ac.uk/engineering/research/mechatronics/constrobotics/main.html

SOURCES

Tiny **'486DX,** Tiny **'486Slc,** VGA104 5SD display card
Advanced Micro Peripherals Ltd.
St. John's, Little Ouse
Nr. Littleport, Cambs
UK CB7 4TG
+44 1353 676311
Fax: +44 1353 6763 13
support@amp-ltd.compulink.co.uk
http://www.controlled.com/pc104/advap1.html

DM5400 ADC I/O card, DM5604 DAC I/O card
Real Time Devices, Inc.
200 Innovation Blvd.
State College, PA 16804.0906
(814) 234-8087
Fax: (814) 234-5218
http://www.rtdusa.com/

MSMCAN CAN bus card
Digital-Logic AG
Nordstraße 4F
CH-4542 Luterbach
Switzerland
+41 32681 53 33
Fax: +41 32 681 53 31
diglogic@spectraweb.ch
http://www.spectraweb.ch/diglogic/

Power Distribution I and II
The parvus Corp.
1214 Wilmington Ave.
Salt Lake City, UT
(801) 483-1533
Fax: (801) 483-1523

M-Systems **2-MB** flash disk
M-Systems
4655 Old Ironsides Dr.
Santa Clara, CA 95054
(408) 654-5820
Fax: (408) 654-9107

7400msi GPS receiver
Trimble Navigation Ltd.
645 N. Mary Ave.
Sunnyvale, CA 94086
(408) 48 I-8000
Fax: (408) 48 I-7744

Satellite-2ASx radio data modem
XL Systems Limited
High Warren House, Tandridge Rd.
Warlingham, Surrey
UK CR6 9LS
+44 1883 622778
Fax: +44 1883 62699 1

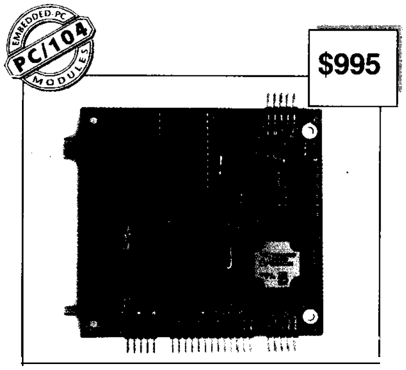
Rotoscan **RS3** optical distance sensor
Leuze Mayser Electronic Ltd.
Alington Rd., Eynesbury
St. Neots, Cambs
UK PE19 2RD
+44 1480403900
Fax: +44 1480 403808

IRS

422 Very Useful
423 Moderately Useful
424 Not Useful

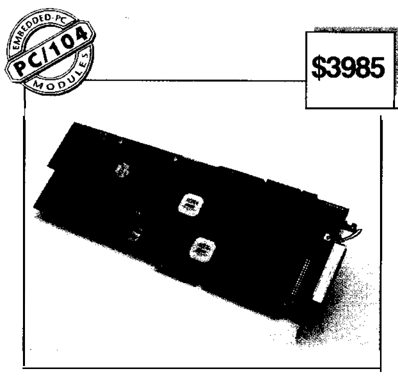
®
rtd
Sets The Pace

Am5x86™ 133 MHz PC/I 04 cpuModules Exceed Pentium-75 Performance



The **CMV586DX133** offers versatile embedded functionality

ISA Bus Intelligent Data Acquisition and Control Boards Multiply Your PC's Processing Power



The IDAC5210 with PC1104 extension bus, 500 KHz analog and digital front-end is powered by on-board **Am5x86** 133MHz CPU and **TMS320C50** DSP

Our PC/1 04 and ISA Bus product lines feature Analog and Digital I/O, CPU, DSP, **Shared Memory, SVGA, PCMCIA, CAN Bus and Intelligent GPS**

rtd® **Real Time Devices USA**
200 **Innovation Boulevard . P.O. Box 906**
State College, PA 16804-0906 USA
Tel: 1 (814) 234-8087 . Fax: 1 (814) 234-5218
FaxBack®: 1 (814) 235-1260
www.rtdusa.com . E-Mail:sales@rtdusa.com

RTD Europa Budapest, Hungary Fax: 36-1-326-6737
RTD Scandinavia Helsinki, Finland Fax: 358-9-346-4539

RTD is a founder of the PC/104 Consortium and the world's leading supplier of intelligent ISA DAS interfaces.

#209

Applied PCs

Fred Eady

A Sweet Solution

Networking Embedded and Desktop PCs Using QNX

This month, Fred turns matchmaker. Teknor's VIPer and a desktop PC meet QNX and 10BaseT. It was love at first sight. Each machine's memory, drive, and port resources became part of the other. Oh! To find such a catch.

It's 10:00 A.M., and I just rigged power to the newest embedded PC in the shop. No smoke! (You weren't expecting smoke, were you? Hey, I'm a professional!)

Normally, "no smoke" signals the beginning of another journey into "Freddy's EPC land." But, this coming-out party is taking a different twist.

You'll recall that I've travelled extensively in the machine-shop world. As a result, I've made many friends.

After all, in my area of the country, embedded PCs control the mills cranking out bunches of mechanical parts that regularly take rides on the birds of NASA's Orbiter fleet. These space-shot machinists occasionally drop in to peek at what's stirring in Circuit Cellar's Florida Room.

This morning, my friend Mark the Machinist is looking over my shoulder. (Mark created my industrial positioner in INK 73.) So far, everything's normal. The familiar BIOS banner appears with the now de facto memory-count screen followed by the "what's in the box" screen.

The memory count reaches 16 MB, and the peripherals include the ever-present floppy drive, a 601-MB Seagate IDE hard file, and the usual assortment of serial and parallel ports. It even has a mouse!

"Oh, neat! An embedded Windows application or one really big ugly DOS project?" Mark quipped. He thinks it's just another of my little embedded marvels.

However, his "All this stuff is running off that little ole board?" marked his jealousy. His lips drooled as "100-MHz CPU Speed" and "Upgradable Flash BIOS" came up.

I giggled as the hardware boot process ended with "Missing operating system".

Thinking himself a savior, Mark proffered, "I'll get the DOS boot disk so you can init the drive."

"No worries. I've got the disk." He paid little attention as I placed the diskette into the drive.

A moment later, "What's this 'Press ESC to boot alternate OS' stuff?" I didn't utter a word as the display glowed with "Welcome to QNX 4.23" and a shell prompt.

Mark pulled up his copilot chair. He knew we were before a truly powerful piece of hardware that was about to be melded with an equally powerful OS.

As we began to explore, it was obvious to us that QNX users, like their UNIX kin, are partial to fish. They always talk about spawning and using forks. We thought about stopping for some good Florida seafood. But not today. There's work to do.

We're installing QNX 4.23 on a Teknor VIPer806 and QNX 4.22 on one of the "more well-equipped" nonembedded demons. Then, we'll tie the two nodes together on a 10BaseT Ethernet network.

The good news is we're taking you along, too. Even if you're a heavy DOS or Windows user, this journey will interest you.

THAT FIRST LOOK

VIPer stands for Very Integrated Processor. As you probably figured, VIPer806 is a complete embedded system on a board. Mine has an Intel '486DX4 running at 100 MHz. (There's a 5x86 133-MHz VIPer variant, too.) Not only does this little reptile hiss, it has big teeth!

I liken the '486-laden VIPer806 to a well-designed, high-tech military aircraft. It supports mission-specific peripheral payloads in all kinds of weather-"all kinds of weather"? Oooh...too many episodes of Wings on the Discovery Channel.

For instance, in hostile industrial environments, the VIPer806 can be configured to operate entirely on flash EPROM, eliminating faults incurred by mechanical drives. In addition, it can operate stand alone or in a passive-backplane system.

The VIPer806 is also capable of carrier launch and recovery anytime, day or night. Gotta see this baby. Check the Table of Contents of this Embedded PC (page 49).

Although the onboard-peripheral firepower is truly awesome, an impressive array of hard and floppy disks, serial and parallel devices, built-in networking, and plug-and-play compatibility can be wing mounted and brought to bear on the most obstinate applications.

The VIPer806 is a complete embedded PC down to the battery-backed real-time clock. In fact, every standard peripheral (and then some) found on the industry-standard PC platforms in my shop can be found on the VIPer806!

Although I received my VIPer with a rudimentary QNX system preloaded in flash, I chose to outfit mine with standard outboard peripherals to demonstrate the extensible power and ease of use that comes from the VIPer806/QNX combination.

AN IMAGE OF SIMPLICITY

Operating systems are born to control their system resources. But, depending on the particular OS-versus-application combination, sometimes "in control of" becomes "out of control with."

Not with QNX. It's a POSIX-compliant OS that controls system resources just like any other well-behaved OS. It looks and feels like any other UNIX system and controls the chaos under the covers in a highly logical and efficient manner.

QNX is well-suited to real-time applications because it provides multitasking and priority-driven preemptive scheduling. It can be configured for a single user or hundreds of users. There's no waste.

Since QNX is process based, you need only include the processes necessary for your particular application. This feature is made possible by QNX's microkernel and message-based interprocess communication architecture.

The core set of QNX documentation including the OS system architecture, user's guide, and utilities reference is exactly 2.92" deep (after bookmark removal). The total documentation measures 2' and takes up my whole bookshelf. It's obvious to the most casual observer that a lot of stuff can be done with the QNX OS! Let's take a peek at its basic system architecture.

MORE THAN SKIN DEEP

Whether you realize it or not, willingly or under duress, you have at some time written a kernel. If you're like me (and you're reading INK, so you are), you tend to cram every conceivable function into your kernel so you don't have to write extra stuff into your application.

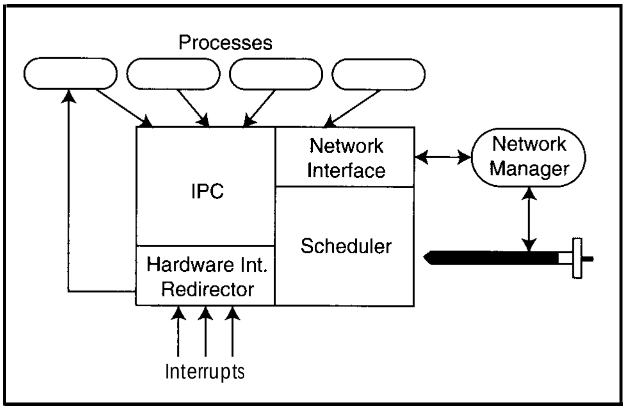


***Figure 1:** Here's a picture of the outfit's brains. As you can ascertain from the diagram, the QNX microkernel is tiny but efficient.*

Before you know it, your little kernel is becoming an OS. Not so with QNX. The QNX kernel is tiny (8 KB to a little over 10 KB, depending on versions) and has only two missions in life-message passing and scheduling.

The kernel itself is never scheduled. Instead, it responds to hardware interrupts and kernel calls. Figure 1 shows the kernel and its relationship to processes.

I'll talk more about processes when I install the OS on the VIPer806. For now, understand that QNX system processes are much like processes you write to make up your application.

In fact, there's very little, if any, difference between a process and an application in QNX. There are no hidden process-interface gotchas. It's like my mom says, "Be careful what you wish for-you might get it."

COMMUNICATION HAPPENS

In a real-world QNX system, the kernel must be able to react to multiple, concurrently executing processes. The most efficientwaytodo thisistoallowtheprocesses to communicate with each other by simply passing messages-something QNX calls IPC (interprocess communication).

In this area, QNX is the pioneer, as far as operating systems go. A process can send, receive, and reply to a message. The messages are not modified by the kernel as they are routed between processes.

Instead, the kernel monitors thechanges in message-process

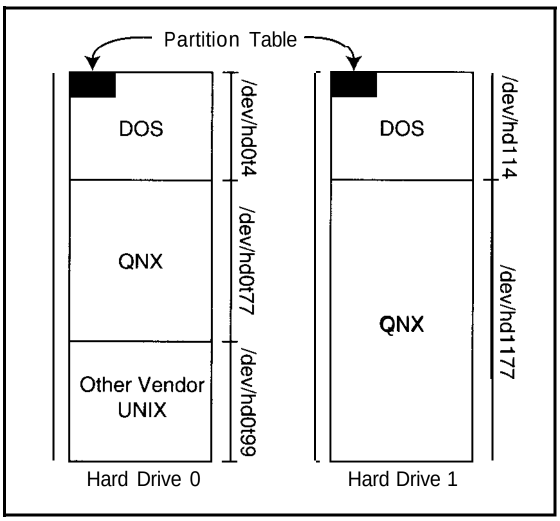
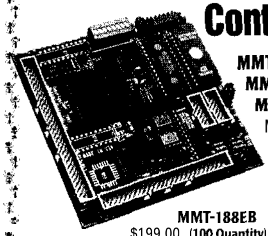


Figure 2: I still think they should call *the* disks with mounted *partitions "cooked." Hard drive 1 is the hard drive 0 setup on my PC in the shop.*

low Cost Embedded
Controllers



MMT-188EB
MMT-Z180
MMT-HC11
MMT-196
MMT-31
MMT-251
MMT-PIC44
&OTHERS
MMT-188EB
$199.00 (100 Quantity)

If you're interested in getting the most out of your project, put the most into it. *Call or* Fax us for complete data sheets and CPU options. Custom work welcome.

MIDWEST MICRO-TEK
2308 East Sixth Street
Brookings, SD 57006
Phone 605.697.8521
Fax 605.692.5112
www.midwestmicro.tek.com
Guaranteed to be in your Future

#210

Listing 1: You could issue `buildqnx build/hard.2 images/hard.2` *or* `make b=hard.2` *to use this file to build the boot image* `hard.2`*. The numbers following the* `$` *are heap size allocations in kilobytes.*

```
sys/Proc32
$ 52000 Proc32 -l 2

sys/Slib16
$ 1 Slib16

sys/Slib32
$ 1 Slib32

/bin/Fsys
$ 8000 Fsys

/bin/Fsys.ata
$ 1000 Fsys.ata

/bin/mount
$ 1000 mount -p /dev/hd0 /dev/hd0t77 /

/bin/sinit
$ 1000 sinit TERM=qnx
```

state and schedules resources accordingly. The changes in state indicate the lifetime of a process. There's also a built-in QNX mechanism synchronizing the message flow between processes.

WHAT'S MINE IS YOURS

The IPC philosophy holds for the entire QNX network. Any QNX node on the network with proper permission can have its processes access any resource on any other node in the QNX network.

There are no local or remote boundaries as far as resources are concerned, and no special application code needs to be written to invoke this functionality. It becomes a vast resource pool that can be channeled as needed to any process in the network.

This form of priority-driven operation inherently puts the QNX network resources where they're needed most. And, IPC is transparent to the point that special code is needed to determine whether a resource is local or on the network.

If it sounds too good to be true, loop on this paragraph until your WOW! flag is set. Now that we've got that straight, let's install some QNX images.

```
PC Side                 VIPer Side
RJ45 Pin                RJ45 Pin
1 <-- [TX+]- - -[RX+]. --> 3
2 <-- [TX-] -- -[RX-]. --> 6
3 <-- [RX+]- - -[TX+]. --> 2
6 <-- [RX-]- - -[TX-]. --> 1
```

***Figure 3:** Hmmm.... looks real similar to something out of an RS-232 null-modem diagram. I told you communications stuff was easy!*

SETTING DATES

Meanwhile, back at the ranch, Mark and Fred are poised at the VIPer806/QNX console. I type `install`, and we're off.

The next screen explains that the install process is done via standard QNX utilities and not some preconcocted install program. Refreshing!

This screen also suggests that DOS be installed first if it's required on the same hard disk. In regards to the VIPer, I ponder the DOS option a moment. Heck, this is a QNX article. I press Enter.

An EIDE controller is found onboard the VIPer806. We're on our way! (I did install a DOS partition on the PC. I'll tell you why later.)

First, QNX starts a conversation with the installed hard disk by starting the Filesystem Manager (`Fsys &`) and the hard disk driver (`Fsys.ata &`). QNX provides drivers for many hard-drive types. Here, the `.ata` signifies that my hard disk has a standard ATA interface.

The Filesystem Manager is responsible for managing block special files. Within QNX, block special files define physical devices like tapes, disks, and disk partitions.

Thus, every disk, tape drive, or partition on a QNX system is known to QNX as a block special file. As the name implies, these disks are viewed within the filesystem as sequential blocks of 512 bytes each. Each block is numbered (beginning with block 1) and extends to the capacity of the physical device.

Once Filesystem Manager and the device driver positively identify the hard drive,

AMX The Real-Time Multitasking Kernel

680x0, 683xx
80386 protected mode
80x86/88 real mode
PowerPC™ family
i960® family
R3000, LR33xxx

NEW: • PowerPC support • *KwikLook* AMX-aware debug tool

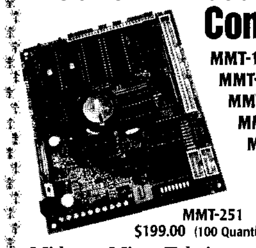
- Compact, ROMable, fast interrupt response
- Preemptive, priority based task scheduler
- Mailbox, semaphore, resource, event, list, buffer and memory managers
- Configuration Builder utility
- Comprehensive documentation
- No royalties, source code included

For a sample of KwikLook and description of AMX,
Phone: *(604) 734-2796*
Fax: *(604) 734-8114*
E-mail: *amxsales@kadak.com*
Web: *http://www.kadak.com*

KADAK Products Ltd.
206 - 1847 West Broadway
Vancouver, BC, Canada V6l Y5

#211

80251 Embedded
Controllers



MMT-188EB
MMT-Z180
MMT-HC11
MMT-196
MMT-31
MMT-251
MMT-PIC44
& OTHERS
MMT-251
$199.00 (100 Quantity)

Midwest Micro-Tek is proud *to* offer its newest line of controllers based on the 8031/51/251 architecture. The 8031 comes in at a surprisingly low cost of $89.00 (100 quantity). Custom work welcome.

MIDWEST MICRO-TEK
2308 East Sixth Street
Brookings, SD 57006
Phone 605.697.8521
Fax 605.692.5112
www.midwestmicro-tek.com
Guaranteed to be in your Future

#212

Mark and I partition the hard disk using QNX `fdisk` (it's very similar to its DOS counterpart).

In fact, the partition information is pretty much identical to the DOS partition information and is kept on the first physical block of the disk. Using the interactive `fdisk` window, I partition the entire drive as a QNX partition. The command to invoke the `fdisk` utility with a "raw" hard disk designated as `hd0` looks like:

```
fdisk /dev/hd0
```

Once the partition is created, the install program mounts it, initializes the filesystem, and mounts the QNX filesystem on the hard disk:

```
mount -p /dev/hd0
```

The `-p` tells the mount utility to read the partition table from the block special file, `/dev/hd0`, and mount all partitions found. Partitions appear as block special files in the format:

```
block special file t partition-
    type-num
```

My QNX partition happens to be type 77, so I end up with:

```
/dev/hd0t77
```

QNX uses a partition number scheme to identify partition types. For example, a 16-bit FAT DOS partition is `/dev/hd0t4`. You can even have another QNX partition on the same physical drive. Here, that partition is `/dev/hd0t78`.

A UNIX partition on hd1 is designated `/dev/hd1t99` and so on. Figure 2 offers a clear view of how QNX partitions are structured.

Next, I initialize the formatted hard disk for use as a QNX filesystem via a utility aptly called `dinit`. (I love this operating system. It's so logical.) It is invoked as:

```
dinit -h /dev/hd0t77
```

Finally, I mount the newly initialized partition at mount point /hd and check it for correct formatting and bad blocks. The syntax is:

```
mount /dev/hd0t77 /hd
dcheck -m /dev/hd0t77
```

The `m` option tells `dcheck` to mark the bad blocks as unavailable.

All the ground work is done, and the Seagate checked out OK. To complete this segment of the installation, `install` copies a QNX license to the hard disk and transfers the rest of the archived operating system files to the QNX partition.

THE RING THING

Everything in its proper directory and linked? Then, it's time to create a QNX boot image and give this machine a node ID for networking purposes. Since I'm concurrently installing a QNX node 1 in the shop (on the PC), the next logical choice for the VIPer806 node is 2.

The QNX boot image is built using the `buildqnx` utility. The boot image is determined by the contents of a build file located in the `/boot/build` directory.

On the VIPer806, this file is defined as `/boot/build/hard.2`. On the PC (node 1), it's `/boot/build/hard.1`.

Invoking the make utility against hard.2 creates a boot-image file in the /boot/images directory. A QNX boot image is simply a collection of processes packaged into a single file.

This single-file image loads into memory at boottime, and control passes to the first process within the image. Normally, the first process starts the remaining required processes. The VIPer806 `/boot/build/hard.2` file contents are in Listing

When QNX boots, the first process in the boot image is always the Process Manager, which contains the QNX microkernel. Processor initialization and process scheduling within the image are boot-time responsibilities of the Process Manager.

The `sinit` utility is the last process in the boot image. It starts a shell that allows the execution of commands from a file.

That command file in the boot instance is the `sysinit.node` file. The `sysinit` file contains commands to initiate service processes for the machine.

Listing 2 is the `sysinit` file for the VIPer806 node. Let's look at it from the top.

The `seti` command establishes some key environment variables (e.g., node num-

IT'S A NEW YEAR!
SUBSCRIBE TODAY AND SAVE
33% OFF THE DOMESTIC NEWSSTAND PRICE
Upcoming issues will feature:
March: Home and Building Automation
April: Robotics
May: Embedded Processors
June: Communications
PLUS our always-popular BONUS SECTION
covering the EMBEDDED PC market
One year (12 issues) for only $21.95 (U.S.)
$31.95 Canada & Mexico, $48.85 all other non-U.S. addresses
(U.S. funds drawn on U.S. banks only)
CIRCUIT CELLAR INK®
THE COMPUTER APPLICATIONS JOURNAL
IT'S EASY TO SUBSCRIBE!
Tel: (860) 875-2199 • Fax: (860) 871-0411 • BBS: (860) 871-1988
Visit our web site at: http://www.circellar.com/

ber and file pathing). TZ in the second line of letclconfiglsysinit.2 defines the time-zone rules such as when to switch to Daylight Saving Time or vice versa. While we're in the zone, rtc 1 hw gets the date and time from the hardware clock.

The next five lines, beginning with Dev &, start the Device Manager. Once the Device Manager starts, the following command lines register the console and serial and parallel I/O with the Device Manager.

Starting with mousetrap start, the command lines up to freeze are pretty obvious. Mouse support is engaged, the floppy system is established, a simple form of IPC called piping is installed, and software floating-point emulation is started.

QNX is smart enough to look for floating point hardware and "slays" (mean-business OS, huh?) the software emulator if the hardware exists.

The Dosf sys utility **enables** data transfers between QNX and DOS partitions. I used it to transfer some listings to a DOS-formatted floppy so I could transmit them electronically to the INK editorial staff.

***Listing** 2: If if ain't here, if ain't **gonna** run! The v **the** Net line helps troubleshoot*

```
-et
export TZ=est05edt04,M4.1.0/2,M10.5.0/2
rtc -l hw
Dev &
Dev.con -n 6 &
reopen //0/dev/con1
Dev.ser &
Dev.par &
mousetrap start
Fsys.floppy &
Pipe &
emu87 &
Dosfsys &
freeze -cdz /etc/logo.F
tinit -T /dev/con* -t /dev/con1 &
Net -v &
Net.ether2100  -d 5 -i 9 -p 300 &
netmap
```

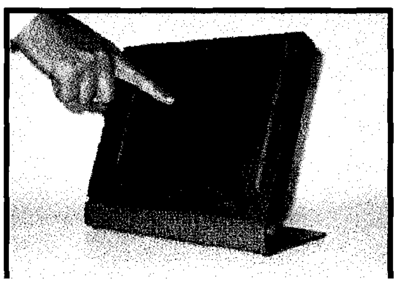
And, this is also why I installed a small DOS partition on the PC. With the DOS partitionand Dosfsys, a Windows-based ftp application can get info and updates from the QNX Web site in QNX archive format ond transfer the downloaded data between the DOS and QNX partitions.

If that's not enough, Dosfsys lets some QNX commands manipulate DOS resources just like DOS would! I like it!

I get a kick out of the freeze command. It's a QNX way of compressing and decompressing (melting) files. It zips and unzips like DOS and Windows but with a logical nomenclature. The QNX logo fades in and freezes into a full screen view.

The login is brought up by the tinit command string. Although tinit can bring up other programs, it is rarely, if ever, seen outside the sysinit environment.

Touch The Future



LCD Touch Monitors
LCD Touch Screens
VGA LCD Displays
LCD Controllers
ISA, PC104, Analog, Video

@EARTH Computer Technologies — Lowest Prices on Earth!

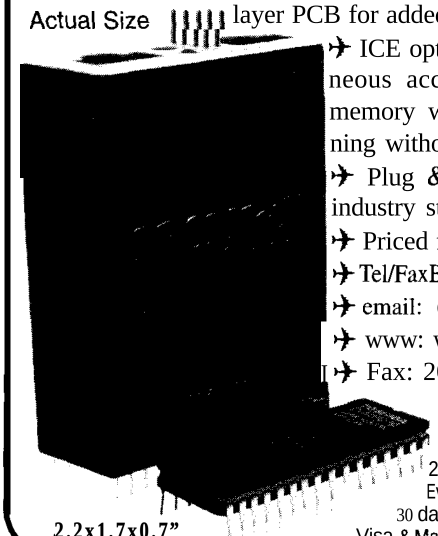
27101 Aliso Creek Rd - # 154 - Aliso Viejo - CA - 92656
Ph: 714-448-9368 - Fax: 714-448-9316
Email: oemsales@flat-panel.com
FREE CATALOG available at http://www.flat-panel.com

#213

PROMJet®-ICE

✈ Ultra compact EPROM and FLASH emulator with high.-st download speed (l-4 Mb/S), largest memory capacity l-32Mb) and fastest access time (85~25ns) in the industry.
✈ Other features include 3V target support, jumperless con-iguration, battery backup, 128 bit bus support and external ower supply. ✈ Fits directly into memory socket or uses xtension cable for flexibility. ✈ Compact design based on igh density FPGAs and double-sided surface-mounted 10 layer PCB for added reliable operation.

Actual Size



✈ ICE option allows simultaneous access to PROMJet's memory while target is running without waitstate signal.
✈ Plug & Play drivers for industry standard debuggers.
✈ Priced from U$295 / MBit.
✈ Tel/FaxBack: 206.337.0857
✈ email: emutec@emutec.com
✈ www: www.emutec.com
✈ Fax: 206.337.3283

EmuTec Inc
Everett Mutual Tower
2707 Colby Av, Suite 901
Everett, WA 98201, USA
30 day money-back policy
Visa & MasterCard accepted

2.2x1.7x0.7"

#214

SID	PID	PROGRAM	PRI	STATE	BLK	CODE	DATA
–	–	Microkernel	–	–		10623	0
0	1	sys1Proc32	30f	READY		106K	233K
0	3	sys1Slib32	10r	RECV	0	57K	4096
	4	/bin/Fsys	22r	RECV	0	81K	2277K
	5	/bin/Fsys.ata	22r	RECV	0	20K	12K
	8	idle	0r	READY	–	0	12K
	15	//2/bin/Dev	24f	RECV	0	28K	53K
	18	//2/bin/Dev.ansi	20r	RECV	0	32K	81K
	20	//2/bin/Dev.ser	20r	RECV		16K	12K
	21	//2/bin/Dev.par	90	RECV		12K	8192
	24	//2/bin/Fsys.floppy	100	RECV		20K	32K
	26	//2/bin/Pipe	10r	RECV		16K	20K
	29	//2/bin/Dosfsys	100	RECV	0	45K	53K
	32	//2/bin/tinit	100	WAIT	-1	20K	16K
	33	//2/bin/Net	23r	RECV	0	24K	81K
	35	//2/bin/Net.ether2100	20r	RECV	0	24K	20K
	37	//2/bin/ksh	100	WAIT	-1	94K	36K
	57	//Z/bin/sin	100	REPLY	1	45K	32K

Figure 4: Here's *the* result *of* executing the sin (display system information) command. Can you find the floating-point emulator *process? I* hope not! Our *VIPer806* is a *'486DX4.*

The final three lines of the /etc/config/sysinit.2 file initiate the Network Manager and start the driver for the network components. In this configuration, the VIPer806 uses the AMD Am79C961 to implement a 10BaseT Ethernet interface.

Net.ether2100 is the driver that supports the AMD device. On the network's PC side, the Net.ether1000 driver supports same fashion. The d, -i, and -p parameters select DMA, interrupt, and port values for the VIPer806 Ethernet interface.

At this point, there's a login prompt on the console. Assuming that a similar process took place on the network's PC side, I can concentrate on what I came for.

I want to tie our freshly QNX'ed embed-

ARRANGING THE DETAILS

If you think I'll need a few paragraphs to walk through building the QNX network, too bad. This ain't no DOS, this ain't no Windows, and this ain't no foolin' around.

Believe it or not, I've already built the network. QNX doesn't require network layers over OS layers. Networking is inherent. Once the hardware network interfaces are installed and the Network Manager finds them, just interconnect the machines.

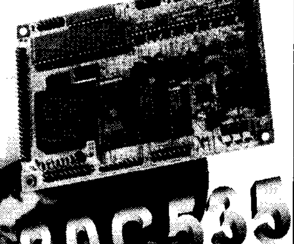
Normally, a UTP-based network of more than a couple of machines requires a hub for operation. Since I'm only putting two machines on my network, I'll use the "null Ethernet" method.

Basically, there are two sets of signals on a 10BaseT backbone-the transmit-data pair and receive-data pair. To effect the "null Ethernet," I simply cross connect the lines as shown in Figure 3.

I'M IN LUV

Bing! Bang! Boom! By establishing a simple QNX network, I've just added almost 1 GB of hard disk, a CD-ROM drive,

$129.95 QTY. 1
SINGLE BOARD COMPUTER



That's right! $129.95 for a full featured Single Board Computer. This board comes *ready to run* featuring the 80535 Processor which is **8051** Code Compatible. Add a Keypad and an LCD and you have a stand alone controller with Analog and Digital I/O.

80C535

* Up to 24 Programmable I/O lines
* 8 Channels of Fast 8/10 bit A/D
* Optional 4 Channel, 8 bit D/A
* Up to 4, 16 bit Timer/Counters with PWM & Watchdog
* Up to 3 RS232/485 Serial Ports
* Backlit capable LCD Interface
* Optional 16 Key, Keypad & Interface
* 192K of Memory Space, 64K included plus EEPROM
* 8051 Assembler & Monitor included, BASIC Opt.

1985 - 1996 OVER 11 YEARS OF SERVICE

EMAC, inc.

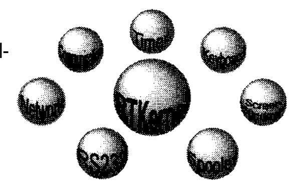
618-529-4525 **Fax 457-0110** BBS **529-5708**
11 EMAC WAY, CARBONDALE, IL 62901

#215

Real-Time Multitasking Tools for Embedded Systems and DOS

RTKernel

Professional, high-performance real-time multitasking system for **DOS** and 16-bit Embedded Systems.
For Borland C/C++, Microsoft C/C++, and Borland Pascal.
Libraries: $550 Source Code: add $500



RTTarget-32

Cross Development System for 32-bit Embedded Systems.
Supports Intel 386 and higher, as little as 16k RAM/ROM.
For Borland C/C++, Microsoft C/C++, and Watcom C/C++.
Libraries: **$1700** Source Code: add $1000

RTKernel-32

Professional, high-performance real-time multitasking system for 32-bit Embedded Systems.
Supports Intel 386 and higher.
For Borland C/C++, Microsoft C/C++, and Watcom C/C++.
Libraries: **$1950** Source Code: add **$1650**

In North America, please contact:
On Time
88 Christian Avenue, Setauket, NY 11733, USA
Phone (516) 689-6654
Telefax (516) 689-1172
email info@on-time.com

Other Countries:
On Time Informatik GmbH
Hofweg 49, 22085 Hamburg, GERMANY
Phone +49-40-2279405
Fax +49-40-2279263
email 102212.3101@compuserve.com

NO ROYALTIES! FREE DEMO DISK!
http://www.on-time.com

On Time
REAL-TIME AND SYSTEM SOFTWARE

two floppies, two serial ports, and two parallel ports to my VIPer806.

Conversely, my PC has come into a little less than 600 MB of extra hard-drive space, a floppy, two serial ports, and a parallel port. All this from an OS called QNX and stringing some wire.

Nothing else to do but logon as root and go after it. Accessing the PC's resources from the VIPer806 is effected by prefixing the request with //1, denoting node 1.

The same (i.e., //2) gets resources on the VIPer from the PC. For instance, //2/etc/readme/tech notes/Net.ether1000 accesses the Net.ether1000 file on the VIPer from the PC.

Changing //2 to //1 does the same from the VIPer. Figure 4 captures the VIPer806 processes running along with their priority, state, and code size.

IT'S FOR LIFE

I'm not done with QNX. There's still over a foot of documentation left to explore!

It's armed with an integral C compiler from WATCOM. And, there's a new version of the Photon microGUI. It's a relatively small footprint-perfect for embedded systems.

There's also TCP/IP support and a utility to run DOS programs under QNX! As they fit, look for me to apply the remaining modules in future projects.

Technical support during the production of this article was excellent. Their Web site is an great place to start with questions or problems concerning an install.

Before I go, there's one gotcha I ran into that put my face in the dirt. You must install a different license of QNX on each node. Otherwise, you'll have the meter out checking perfectly good network cables while your nodes are laughing at you. APC.EPC

Special thanks to Tracy Nicholson at QNX customer support for slaying all the dragons that stumbled into my QNX kingdom.

Fred Eady has *over 19 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

T. Green and C. Scott, QNX OS *VEDIT* Text *Editor,* QNX Software Systems, V.2.10, April 1995.

QNX Software Systems, *QNX* OS System *Architecture,* February 1995.

QNX Software Systems, *QNX* OS User's Guide, April 1995.

QNX Software Systems, *QNX* OS *Utilities* **Reference,** April 1995.

Teknor Industrial Computers, *VIPer806* '486/5x86 Single *Board* **Computer** Technical Reference *Manual,* V.2.2, August 1996.

SOURCES

QNX OS

QNX Software Systems Ltd.
175 Terence Matthews Crescent
Kanata, ON
Canada K2M 1W8
(613) 591.0931
Fax: (613) 591.3579
http://www.qnx.com/

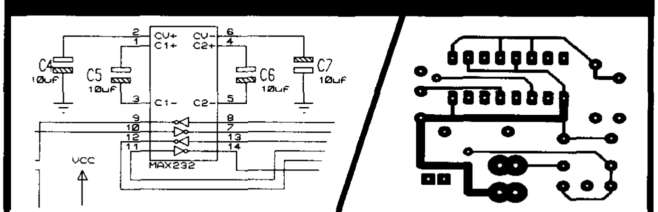
VIPer806

Teknor Industrial Computers, Inc
616 Cure Boivin
Boisbriand, PQ
Canada J7G 2A7
(5 14) 437-5682
Fax: (5 14) 437-8053

IRS

425 Very Useful
426 Moderately Useful
427 Not Useful

EASY TO USE CAD



Now in Both Windows and DOS

CAD PA K *Windows-$199, DOS - $159*

Ideal for New Users, Hobbyists, & Small Businesses!
Provides everthing for PCB Layout & Schematic Drawing

PROTEU S *Starts at $425*

Most Powerful EDS System on the market!!!

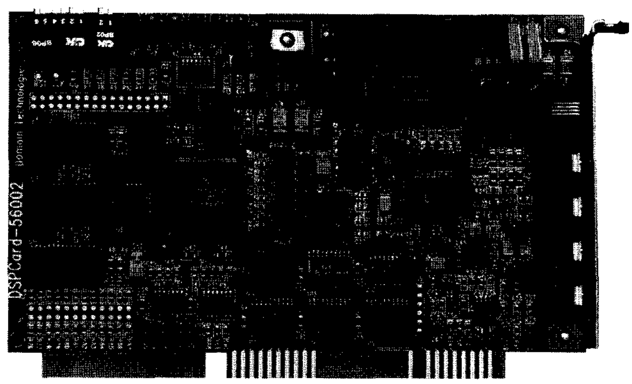
* Schematic Capature * Circuit Simulation
* PCB Layout with Rip-Up & Retry Autorouting

ALL PRICES ARE IN US FUNDS, PLS INCLUDE $10 S/H

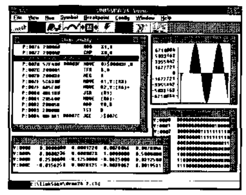
R4 SYSTEMS INC.
1100 GORHAM ST. Suite 11B-332
NEWMARKET ONTARIO
CANADA L3Y 7V1
905 898-0665 FAX 905 898-0683
BBS 905 898-0508 (9600,8,N,1)

FREE DEMO
WRITE OR CALL
TODAY

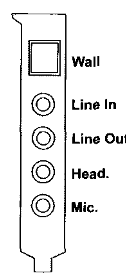
Download Demo - http://www.r4systems.on.ca
Internet email info@r4systems.on.ca



Motorola 56002 DSP Card



General Purpose Card
66 MHz, 24-bit 56002
Stereo Sound Port
Telephone Port
128 KW, 15 ns SRAM
2 Interfaces with the PC
Supplied with Software
Symbolic Debugger
Assembler by Motorola
$800



Domain Technologies, Inc.
Tel: (972) 578-1121
Fax: (972) 578-1086
Email: info@domaintec.com
http:// www.domaintec.com

DOMAIN TECHNOLOGIES, INC.

#216

FROM THE BENCH

Jeff Bachiochi

Unconventional Access

Sometimes you inherit a project and you can't get to the code. If that's your situation, then Jeff has a BUD for you. No, not the kind you drink. He offers a BASIC untokenizing device. Pop this one open, and pretty soon, you'll be in business.

Picture this. It's Saturday evening, and the family is crowded around the small-screen TV. There's absolutely nothing on (IMHO). Maybe a field trip would help.

I needed to pick up some ceiling paint to repair the water stains caused by ice backing up on the roof last winter. Off we went to Home Depot.

I picked up double-sided tape to fix the blinds, caulk to fix the countertop, and other hardware items. Bev grabbed some heavy-duty spray cleaner for the water spots in the shower.

While the kids composed a symphony on the door-bell display, my eye was caught by an RF doorbell by Trine Products-a doorbell button transmitter and a "no wires" chime unit. At $15 for the transmitter/receiver pair, I couldn't pass it up.

We finally left to the tune of 80 bucks. All the way home, I couldn't stop thinking about where I might use my new find until Bev asked about the paint. Huh?

STARTING FRESH

Nothing like getting back to work Monday morning to help bring one back to normality-or so it seems. I'm not even through my first cup of coffee, and the phone rings with a plea for help.

A pleasant-sounding gentleman asks what tools he needs to change an existing program presently running in the company's embedded controller. I explain the tools necessary, based on the information he gives me. He says thanks, and I hang up thinking this is going to be a good week.

A few days later, John calls again. "OK," he says, "I've got everything in front of me. How do I get the program out of the controller?"

Right then, I got a bad crick in my neck. Politely, I tell him he can't afford to have me read manuals to him. I provide him with a list of procedures to accomplish his task, ask him to read the manuals, and tell him to call if he gets frustrated.

It didn't take long. He called again that afternoon.

Truth was, he had inherited this project. The former project engineer had left the company, and no documentation could be found.

The program was written in BASIC52. However, the system had no console serial port, so access to the program was impossible.

John not only needed to know how to edit a program file and produce a new ROM but also how to get the original program out of the controller. No offense, but I felt he was in way over his head.

BASIC TO THE RESCUE

When you have access to the console serial port on an embedded controller using BASIC52, you have access to the program (unless some protection schemes are implemented). The BASIC52 processor converts the ASCII data of your BASIC program to a tokenized record stored in RAM.

Whenever you list or edit it through the serial console, the BASIC52 processor converts the tokens back to ASCII data. You never even realize it's tokenized unless you look into the RAM spaces.

When BASIC52 recognizes one of its 110 BASIC commands, it replaces the command with a single byte of data. This way, program space is condensed, and the commands don't have to be reinterpreted to be processed. The commands are given values above

Listing 1—Here's *the GWBASIC* program *created to convert an Intel hex file into a BASIC52 ASCII* **text file.**

```
10 DEFINT A-Z
20 PRINT"This program will prompt for an Intel hex filename."
30 PRINT "If the file is a tokenized BASIC-52 file,"
40 PRINT"an untokenized ASCII file is created."
50 PRINT
60 INPUT"What is the Intel hex filename? ",A$
70 CLS
80 OPEN A$ FOR INPUT AS #1
90 B$=""
100 PRINT
110 FOR A=1 TO LEN(A$)
120 IF MID$(A$,A,1)<>"." THEN B$=B$+MID$(A$,A,1) ELSE GOTO 140
130 NEXT A
140 B$=B$+".BAS"
150 OPEN B$ FOR OUTPUT AS #2
160 PTR=0
170 GOSUB 4000: REM Look for a BASIC52 program fingerprint
180 GOTO 200
190 C$=RIGHT$(C$,LEN(C$)-1)
200 IF LEN(C$)=0 THEN GOSUB 2000: REM Get more raw data
210 IF PTR=0 THEN GOTO 260: REM Line length
220 IF PTR=1 THEN GOTO 270: REM Program line number high byte
230 IF PTR=2 THEN GOTO 280: REM Program line number low byte
240 GOTO 400: REM Program line data
250 GOTO 190: REM Next data byte
260 D$=" ":NUMCHAR=ASC(LEFT$(C$,1)):IF NUMCHAR=1 THEN PRINT "Finished
      processing": END ELSE PTR=1: GOTO 190
270 HB=ASC(LEFT$(C$,1)): PTR=2: GOTO 190
280 LB=ASC(LEFT$(C$,1)): LN=HB*256+LB
290 LOCATE 2,1: PRINT"Writing output file program line #",LN
300 V=INT(LN/10000):IF V>0 THEN D$=D$+CHR$(V+&H30): LN=LN-(V*10000)
310 V=INT(LN/1000):IF V>0 THEN D$=D$+CHR$(V+&H30): LN=LN-(V*1000):GOTO
      330
320 IF LEN(D$)<>0 THEN D$=D$+CHR$(V+&H30): LN=LN-(V*1000)
330 V=INT(LN/100):IF V>0 THEN D$=D$+CHR$(V+&H30): LN=LN-(V*100): GOTO
      350
340 IF LEN(D$)<>0 THEN D$=D$+CHR$(V+&H30): LN=LN-(V*100)
350 V=INT(LN/10):IF V>0 THEN D$=D$+CHR$(V+&H30): LN=LN-(V*10): GOTO
      370
360 IF LEN(D$)<>0 THEN D$=D$+CHR$(V+&H30): LN=LN-(V*10)
370 IF LN>0 THEN D$=D$+CHR$(LN+&H30): GOTO 390
380 IF LEN(D$)<>0 THEN D$=D$+CHR$(LN+&H30)
390 PTR=3: GOTO 190
400 V=ASC(LEFT$(C$,1))
410 PTR=PTR+1
420 IF PTR=NUMCHAR THEN D$=D$+CHR$(V): PRINT#2,D$: PTR=0: GOTO 190
430 IF V<&H80 THEN D$=D$+CHR$(V): GOTO 190
440 ON (INT((V-&H80)/16))+1 GOTO 450,620,790,960,1130,1300,1470,1640
450 ON V-&H7F GOTO 460,470,480,490,500,510,520,530,540,550,560,570,
      580,590,600,610
460 D$=D$+" LET'": GOTO 190
470 D$=D$+" CLEAR": GOTO 190
480 D$=D$+" PUSH ": GOTO 190
490 D$=D$+" GOTO ": GOTO 190
500 D$=D$+" PWM ": GOTO 190
510 D$=D$+" PHO. ": GOTO 190
520 D$=D$+" UI": GOTO 190
530 D$=D$+" UO": GOTO 190
540 D$=D$+" POP ": GOTO 190
550 D$=D$+" PRINT": GOTO 190
560 D$=D$+" CALL ": GOTO 190
570 D$=D$+" DIM ": GOTO 190
580 D$=D$+" STRING ": GOTO 190
590 D$=D$+" BAUD ": GOTO 190
600 D$=D$+" CLOCK": GOTO 190
610 D$=D$+" PH1.": GOTO 190
:
2000 C$="": REM Read and check an Intel hex paragraph
2010 IF EOF(1) THEN PRINT"End of File": STOP
2020 INPUT #1, I$
2030 IF (MID$(I$,1,1)<>":")THEN PRINT"Error, character should be: "
2040 X=2: GOSUB 3000: L=V*16: X=3: GOSUB 3000: L=L+V
2050 X=4: GOSUB 3000: NA=V*4096: X=5: GOSUB 3000: NA=V*256+NA: X=6:
      GOSUB 3000: NA=V*16+NA: X=7: GOSUB 3000: NA=V+NA
2060 X=8: GOSUB 3000: M=V*16: X=9: GOSUB 3000: M=V+M
2070 IF (M<>0 AND M<>1) THEN LOCATE 3,1: PRINT"Mode Error: must be 00
      or 01"
2080 IF (M<>0) THEN LOCATE 3,1: PRINT"Mode: End of File--Conversion
      Complete": CLOSE: END
```

(continued)

80h, leaving 00-7Fh for any 7-bit ASCII character.

Without the serial connection, the processor has no way of delivering an untokenized listing to the user. This became John's predicament.

He suggested removing the ROM and getting it read with an EPROM programmer. After a careful explanation of BASIC52's tokenizing process, I had him totally confused.

Should I give in and tell him to ship me the ROM and I'd send him back a listing? What if further protection was implemented to prevent the code from being reverse engineered?

What if-pardon the accusation, John-this was really a covert operation to illegally copy someone else's protected code? Wouldn't a legal owner have proper documentation?

I quickly put on my marketing hat and convinced him that, for less than $100, he could get a "development" board (with the necessary console serial port on it) and use it to read out the ROM's program. I knew this wasn't the last time he'd call, but at least next time, he'd have the tools he needed for me to talk him through it.

The more I thought of John's predicament, the more I realized how handy it would be to be able to convert a BASIC52 (tokenized) Intel hex file back into native ASCII text. I decided to create the BASIC Untokenizing Device-BUD, for short.

So, John, this BUD's for you!

BASIC PROGRAM STORAGE

On powerup or reset, the BASIC52 processor reads address 8000h. A 1-6 (ASCII 31h–36h) tells it to use one of the six special start-up procedures.

These procedures include a number of options. You can automatically run the program stored at location 8011h or use a baud rate for serial communication based on a timer reload value (RCAP2) located at address 8001–8002h. You can also automatically set the baud rate based on a received character or set an upper RAM boundary based on address 8003–8004h.

The tokenized BASIC program starting at 8011h must have a 55h preceding it at address 8010h to signify that a valid program follows.

BASIC52 program lines are stored in a simple format. The first character is a line-length indicator. (Although the maximum is 255, the actual value is much less since the ASCII line length must be fewer than 80 characters before tokenizing.)

The second and third characters make up the line number (maximum 65535). The fourth and following (through the line length) are either characters (ASCII values <80h) or tokens (ASCII values >7Fh).

If a line starts with a 01h character, there are no more lines (i.e., it's an EOF marker). Otherwise, additional program lines continue as above.

Unless the design of the embedded system is really off the wall, the ROM (which holds this program) can only be in one of two locations-a 64-KB ROM starting at OOOOh (code begins at address 801 1h when read with a programmer) or a smaller ROM starting at 8000h (code begins at address 001 1h when read with a programmer).

So, the hex code read into the programmer and saved to a file may start at address OOOOh or 8000h. It's also possible that the tokenized BASIC52 program may not have any special start-up procedures saved, so it's necessary to check for the 55h before assuming the file is good.

INTEL HEX FORMAT

A file stored in Intel hex format has built-in safeguards against data errors. Each individual line begins with a ":" and ends with a two-character checksum of the whole line to assure data integrity.

However, the file format does have one small gotcha. There is no overall checksum. If a complete line gets lost, there's no way of knowing.

All characters in an Intel hex file are 7-bit ASCII characters which ensures all characters are printable. Two hexadecimal characters combine to create a single 8-bit binary value.

The first pair (following the ":") is a line-length count. The second and third pair combine to make a load address for the first data byte.

The fourth pair indicates a mode byte for the line ("00" means data in the line, and "01" is the EOF marker). If the mode is "00", data follows in pairs up to the checksum byte [the last pair in the line).

STILL CRAZY AFTER ALL THESE YEARS

The PC is my tool. I still use GW-BASIC, which was distributed with DOS prior to QBASIC. It seems to be vanilla enough to run on any PC. The command set is universal as it doesn't use any special functions.

Most beginners can follow and understand the program flow, while experienced programmers can use it as a guide for rewriting the routines in their favorite language. Call me crazy, but I use what works best for me.

One approach might be to read in the whole Intel hex file and construct the binary values into a buffer, so it looks exactly like it does in the embedded controller's memory.

I chose to read the file a line at a time, processing it until more data is needed. This saves on overall memory necessary for the task. Not that I don't have gobs of memory to waste in my PC. Isn't it written somewhere that programmers must use the full extent of available memory?

Individual lines (paragraphs) are read from a user-prompted input file (USERFILE.HEX). Converted program lines are then written to the output file (USERFILE.BAS):

```
10 LET MTOP=1FFFH
20 TIME=O: CLOCK1
30 LET X=0
40 DO: X=X+1
50 PRINT"Hello World",
60 UNTIL X>999
70 PRINT"That's all..."
80 PRINT TIME," seconds"
90 END
```

A line is read into a string variable (I $) and checked for legality. The data from the line is converted into binary and stuck into a second string variable (C $). The program then uses the load address (from the . HEX line in I $) to search the data bytes (up to the data length of C$) for any of the fingerprint information.

It continues collecting until it finds a 55h at either 0010h or 8010h. If the load address exceeds 8010h

Listing I-continued

```
2090 FOR X=10 TO 10+(2*(L-1)) STEP 2
2100 GOSUB 3000: CS=V*16: X=X+1: GOSUB 3000: CS=V+CS: X=X-1
2110 C$=C$+CHR$(CS)
2120 NEXT X
2130 CS=0
2140 FOR X=2 TO 10+(2*L) STEP 2
2150 GOSUB 3000: CS=V*16+CS: X=X+1: GOSUB 3000: CS=V+CS: X=X-1
2160 NFXT X
2170 IF (CS AND &HFF)<>0 THEN PRINT"Checksum Error": STOP
2180 LOCATE 1,1: PRINT"Reading input file line #",ILN: ILN=ILN+1
2190 RETURN
3000 REM ASCII data to binary conversion
3010 IF (MID$(I$,X,1)>="0" AND MID$(I$,X,1)<="9") THEN
     V=ASC(MID$(I$,X,1))-&H30: RETURN
3020 IF (MID$(I$,X,1)>="A" AND MID$(I$,X,1)<="F") THEN
     V=ASC(MID$(I$,X,1))-&H37: RETURN
3030 PRINT "Data Error": RETURN
4000 GOSUB 2000: REM Get data and look for BASIC52 fingerprint
4010 FOR X=NA TO NA+L-1
4020 IF (X>&H10 AND X<32784!) THEN GOTO 4100:REM &h8010
4030 IF (X=0 OR X=32768! AND MID$(C$,X-NA+1,1)>"0" AND MID$(C$,X-
     NA+1,1)<"7") THEN P=ASC(MID$(C$,X-NA+1,1))
4040 IF (X=1 OR X=32769!) THEN IF P<>0 THEN RH=ASC(MID$(C$,X-NA+1,1))
4050 IF (X=2 OR X=32770!) THEN IF P<>0 THEN RL=ASC(MID$(C$,X-NA+1,1))
4060 IF (X=3 OR X=32771!) THEN IF P<>0 THEN BH=ASC(MID$(C$,X-NA+1,1))
4070 IF (X=4 OR X=32772!) THEN IF P<>0 THEN BL=ASC(MID$(C$,X-NA+1,1))
4080 IF (X=&H10 OR X=32784!) THEN IF MID$(C$,X-NA+1,1)=CHR$(&H55) THEN
     S=255
4090 IF S=255 THEN C$=RIGHT$(C$,(L+NA-X-1)): GOTO 4130
4100 NEXT X
4110 IF (X<32778!) THEN GOTO 4000: REM &H8011
4120 PRINT"No BASIC52 found": END
4130 PRINT#2,"REM Prog number = ",HEX$(P)
4140 PRINT#2,"REM RCAP2 = ",HEX$(RH*256+RL)
4150 PRINT#2,"REM MTOP = ",HEX$(BH*256+BL)
4160 RETURN
```

```
:2000000034FFDC7FFFFFFFFFFFFFFFFFFFFFFF550C000A80C4EA3146464646480D0A00144F
:20002000C5EA303A8E310D08001E8058EA300D050028940D13003289224865 6C6C6F20578D
:200040006F726C64222C0D0A003CA258EF3939390D140046892254686174277320616C6C24
:200060002E2E2E220D11005089C52C22207365636F6E6473220D05005AA30D01FFFFFFFF80
:20008000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF80
.
. (lots of empty space)
.
:00000001FF
```

Figure 1—*This* example of an *Intel* hex file was read from a ROM containing *the program in* Listing 1.

without finding a 55h, then the load stops with a "No BASIC found" message. If 55h was found at either load address (0010h or 8010h), the program saves the autostart information to the output file and proceeds with the second phase-untokenizing the data.

Once it's determined there is a BASIC52 program here, program lines can be built and each completed line output to a file. It is possible to have a program indicated when there isn't one, but I assume there is one.

It's important to note that the binary data in C$ does not necessarily start and/or end at the beginning or end of a BASIC52 program line. The data in C$ comes from an Intel hex file and is not presented in chunks equal to a single BASIC52 program line. So, C$ is simply tested prior to removing characters, and another line from the hex file is added whenever the length of string C$ equals zero.

The first character from C$ is the length of program line. The following two bytes equate to the program's line number. A new string (D $) is built beginning with an ASCII representation of the line number.

For the rest of the line length, I look for tokens. If the value of the character is less than 128, then the data is assumed to be an ASCII character and is added to D $. If the value of the character is greater than 127, the data is assumed to be a token.

The program branches first to a page address based on the token value/16 and then to an offset address based on the token-page address. Each token directs the program flow to its own line, where the token is replaced by a special set of characters.

These characters are supposed to be the same characters that the token originally replaced by the 8052 BASIC processor. These characters and not the token are added to string D$.

After reaching the last character (from the length-of-program-line character), the string D$ is written out to disk, and the new length-of-program-line character is read from C$, starting the line conversion over again.

Eventually, an EOF marker is reached either from the Intel hex file (indicating trouble) or the program line-length character 0 1 h (signifying all's well). The input and output files are closed, and operation comes to a halt. And, if the planets are aligned properly, you end up with a file containing the original BASIC program.

Let's run a small test program using BUD (see Listing 1). Figure 1 shows what the tokenized file looks like when read into an EPROM programmer from a 27C64 and saved as an Intel hex file.

Since the EPROM has room for 8 KB of code, most of the file is filled with FFs (i.e., unprogrammed or erased address locations). The program could look for multiple BASIC52 programs within the same file (multiple programs are legal in BASIC52), but that exercise is left to the user.

Here's what the translation looks like by running it through BUD:

```
REM Prog number = 34
REM RCAP2 = FFDC
REM MTOP = 7FFF
10 LET MTOP=1FFFH
20 TIME=0: CLOCK1
30 LET X=0
40 DO: x=x+1
50 PRINT"Hello World".
60 UNTIL X>999
70 PRINT"That's all..."
80 PRINT TIME." seconds"
90 END
```

The REM statements in the beginning show the autostart information also held within the hex file.

The P ROG number is an ASCII 4 (PROG4). The RCAP2 reload value equals 9600 bps (when using an 11.0592-MHz crystal). MTOP (top of RAM) is initially set to 7FFFh.

The mystery program (had we not had a record of it) is no longer a mystery. BUD has decloaked the Romulan "Hello World" program which most of us would prefer to never see again.

Now, I get to add another tool to my virtual toolbox. Oh, no. Looks like I'm gonna need a bigger box.

CAREFUL-YOU MIGHT GET WHAT YOU ASK FOR

Any time I have success with a project [no matter how big or small), I like to celebrate. The celebration might consist of a cold beer, an afternoon of windsurfing, or maybe just a handful of gourmet jellybeans from the candy dish on my desk.

My family's big treat is to go out for supper to Mickey D's. Because my tastes don't include onions, I always special order. "A #6 (two cheeseburgers). Just ketchup and pickles."

Imagine setting your teeth into a cheeseburger and ending up with just ketchup and pickles between the buns. Yup, that's it, no burgers. Back up to the counter I schlep.

"Now I know McDonald's burgers are small," I complained, "but... where's the beef?" Someone said something about it being Wendy's fault. We all had a good chuckle. Just another normal day.

Oh, the RF doorbell. I decided to connect the RF doorbell to the mailbox. Whenever the mailbox is opened, a small micro switch makes contact and out goes an RF transmission.

The receiver located in the kitchen lets us know when the mail arrives. But, you never know when it's just the Publisher's Clearing House.

See you next month. I gotta go get some ceiling paint.

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.*

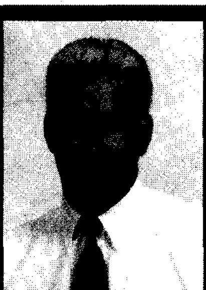
IRS

428 Very Useful
429 Moderately Useful
430 Not Useful

Radar Love

SILICON UPDATE

Tom Cantrell



It's not just little boys that ogle over the idea of x-ray specs. After Tom gives you the full scoop, you will, too. They work like any motion or proximity detector, except they see through walls, ice, solid objects....

Oike me, I'm sure many of you read comic books as a kid. Remember tht ad, typically on the back inside cover, hawking all manner of strange and wondrous (at least to a nine year old) geegaws?

I can still remember scraping my allowance together and making the tough choice between a smoke bomb and the x-ray specs, with the former winning the call.

It seemed to take forever for that small box in a plain brown wrapper to arrive. I snatched a book of matches and headed for the alley, excited and a little scared by the pyracy at hand.

Of course, I ended up going through the whole matchbook trying to get the darn thing to emit more than a fitful smolder. Yeah, it hurt, but I learned some valuable lessons early on.

By now, you (and my long-suffering editor) may be wondering whether I can possibly concoct a relevant high-tech connection with comic-book smoke bombs and x-ray specs. Heck, that's easy.

THE ULTIMATE GARAGE SALE

When thinking of bombs, one of the first places that comes to mind is Lawrence Livermore National Labs. Just so you don't make the mistake attributed to one DOD poobah and ask "Just who was Lawrence Livermore anyway!", you should know LLNL was named after UC Berkeley physicist E.O. Lawrence and is located in the rolling hills of Livermore, just east (er, downwind) of the San Francisco Bay area.

The Lab's genesis can be traced to that most human of instincts-NIH. Fact was, a number of leading bomb experts wanted a second, fully equipped lab to develop thermonuclear weapons in "competition" with Los Alamos. Needless to say, a number of powers that be (including those at Los Alamos) had to chew on that for a while.

Lawrence, a longtime University of California employee, was the one who understood how to get along. The Lab's charter was finessed to read "methods and equipment" in "collaboration" and only later, perhaps, some "broader programs."

He also knew you had to ease into these things to slip past all the attention a budget buster inevitably attracts. Thus, in 1952, the Lab got off to a rather humble start (an early memo laments, "The desk-lamp situation is very bad. Every man needs a desk lamp.") as a mild-mannered branch of the University of California.

These days, with close to 10,000 employees (1000+ Ph.D.s) and a budget approaching $1 billion per year, I presume desk lamps aren't a problem.

However, the bomb-making business being what it is, the Lab is under constant pressure to boost the nondefense portion of their budget. Wondering whether your paycheck will go down in flames as the first casualty of the line-item veto is a great motivator.

So, the call has gone out for the SWAT (Sell What's Available Today) team to search the nooks and crannies and make some deals. In fact, there are quite a few neat goodies rattling around in the attic.

Naturally, many are closely related to bomb stuff, whether building 'em (Superplastic Metal Forming), blowing 'em (Statistical Models of Rock Fracture), or cleaning up (Robotic Handling and Processing of Mixed Wastes). However, there's a bunch of interesting nonbomb stuff, too.

For instance, lasers are a hot item since they exhibit an admirable diversity of military, big science, and commercial potential. And when it comes to lasers, LLNL, home of the football-field-sized, fusion-capable 100-trillion-watt Nova (and proposed site for the even larger, if somewhat ominous sounding, National Ignition Facility), has the right to sneer, "You talking to me?"

Now, it turns out, when you're fooling around with laser fusion, you've got about a billionth of a second to gather your data, so it's not a good time to be fumbling around with your VOM. Anyone who's been shopping around for 1-GHz scopes knows such speedy data-acquisition gear doesn't come cheap.

Thus, it came to pass that a few engineers nestled deep in the bowels of the Lab came up with some clever-and cheap— ways of recording subnanosecond events.

Somewhere along the line, a chap named Tom McKewan realized that there were intriguingly practical product possibilities. Turns out the ability to grab and decipher subnanosecond signals is pretty useful— especially if it only takes a handful of low-tech components

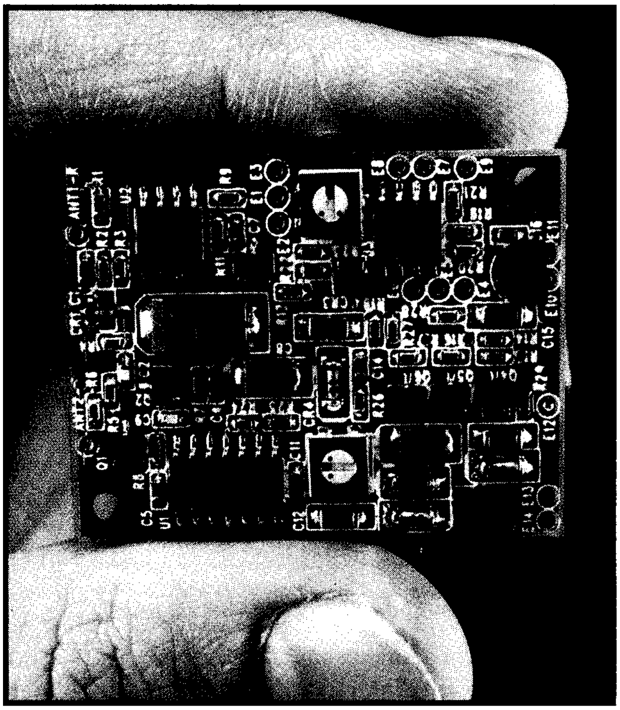


Photo I-M/R (Micropower *Impulse* Radar) *is an* intriguing product from *an* unlikely *source.*

RADAR FOR THE MASSES

One of the most interesting applications to emerge is known as Micropower Impulse Radar (MIR), which does just what it says. As you see in Photo 1, it's a radar that fits in your pocket (and your budget), running off AA batteries.

Taking the radar (Radio Detection and Ranging) refresher in Figure 1, we are reminded that radar works by emitting a radio-frequency (RF) signal and listening for the echo. The range is simply the product of the elapsed time and the speed of light.

The concept is analogous to the ultrasonic (i.e., sonar) ping-and-echo sensors used for proximity detection (i.e., auto-focus cameras, fluid level, electronic tape measure, etc.).

The concept may be the same, but boosting the wavelength from near audio (e.g., 40 kHz) into the RF spectrum calls for some speedy logic (see Table 1). And, to allow battery operation and eliminate medical concerns, the pulses are extremely short (hundreds of picoseconds) and weak (averaging tens of microwatts). To top it all off, there's tons of RF garbage about, including plenty of radio and TV, not to mention garage-door openers, security systems, and our beloved PCs.

To minimize the effects of noise, the MIR uses the range-gate approach. It waits for a certain delay after transmission and then only listens for a short period of time. In other words, the MIR output is basically binary, reflecting the presence or absence of a target at the distance corresponding to the range-gate delay.

Don't expect to see that 200-ps pulse with your friendly neighborhood DSP, since even the whizziest chips aren't anywhere near fast (or cheap) enough. Instead, the MIR exploits a clever low-tech trick, using a bank of capacitors to sample and hold the waveform as it marches in from the antennae.

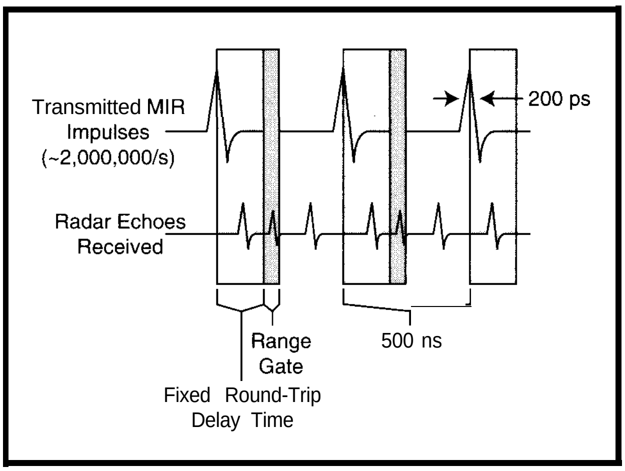


Figure I--The M/R breakthrough *is the ability to* generate *and detect s&nanosecond events cheaply and with little power.*

On this humble foundation, basic MIR operation is enhanced in a number of ways. Simply averaging thousands of samples improves the signal-to-noise ratio and filters much of the clutter. As a further hedge, the MIR adopts spread-spectrum techniques by randomly varying the exact time between pulses, thereby countering continuous interference that might otherwise alias into the radar samples.

Ironically, presuming the MIR biz takes off, a likely source of interference will be another MIR unit. Once again, randomization allows multiple units to operate in proximity, without devoting a particular frequency to each.

Spreading the pulse frequency and the low-output power also gives the MIR stealth-like characteristics (i.e., to an eavesdropper, it's hard to differentiate from background noise). It's perfect for security types, and it doesn't blitz your neighbor's TV, either.

SWORDS INTO SENSORS

The physic basics of MIR dictate certain operating characteristics. For instance, the wavelength dictates the minimum object size and detection range (typically ~6"), while the low power limits the range [tens of meters in air]. Many other specs (e.g., range and directionality) are a function of antenna design, starting with a couple inches of wire and moving up to reflectors, horns, and so forth.

Otherwise, the MIR is like a variety of other motion and proximity sensors, except with the rather radical advantage of being able to see through walls, concrete, mud, ice, and other solid objects.

Antenna pattern (H-plane)	360° with dipole antenna, 160° with cavity-backed monopole and narrower with horn/reflector/lens
Center frequency	1.95 or 6.5 GHz ±10%
Emission bandwidth	500 MHz at 1.95 GHz center
Average emission power	-1 µW (measured)
Duty cycle	<1%
PRF (average)	2 MHz ±20%
PRF coding	Gaussian noise, low coherence swept FM, or pseudonoise
Receiver noise floor	<1 µVrms
Receiver gate width	250 ps for 1.95GHz system
Range delay	RC analog, pot/DAC controllable
Range-delay jitter	<1 ps rms
Range-delay stability	RC component limited over temperature (drift in range delay expands/shrinks shell)
Detection range	adjustable from 2" to 20'
Motion passband	0.3–10 Hz, Doppler-like signature
Analog output	-0.1-2-V peak on motion sensing, hand motion at 6' gives ~300-mV peak
Receiver gain	70 dB
Power	normal mode: 5 V at 8 mA; long battery life: 2.5 V at 20 MA
Size	1.5 in? SMT PCB with 1.5" long wire dipole elements
Semiconductors	74AC00 CMOS (1 each), bipolar or CMOS op-amps (2 each, quads), Si-bipolar RF transistor at >4 GHz ft (2 each), Schottky diodes, Cj (0) <1pF (2 each)

Table I-One *obvious use for MIR is as a motion sensor for security applications.* Compared *to* widely used/R sensors, *the MIR unit offers* hidden mounting, *stealthy* signature, precisely tunable coverage, and *fewer* false alarms.

One obvious candidate for MIR application is as a basic motion sensor for security applications (see Figure 2). The MIR-based sensor can be stuck in a closet, mounted behind a picture, or put in the attic. You can cover your backyard through a window, avoiding the environmental and wiring challenges of outdoor sensors.

Unlike the popular IR motion sensor, MIR achieves precise coverage based on range-gate and antenna design. This precision allows selective detection of, for instance, an adult, child, or pet-a level of discrimination hard to achieve with IR. It should yield a lot fewer false alarms, especially outdoors where it seems like jùst about everything-wind, rain, snow, and all manner of critters-fools JR detectors.

Indeed, MIR's resolution is so high, it can easily detect human respiration from a distance, leading to all kinds of Star Trek-like lifeform detectors. For instance, a cop searching for a bad guy might like a little gadget that warns if someone's hidden in the closet.

Or for disaster response, consider a radar packaged in a frisbee, which can be flung into an area to detect whether someone is buried. Closer to home, how about a baby monitor that eliminates SIDS (Sudden Infant Death Syndrome) once and for all.

By taking multiple measurements as the range gate is swept, the MIR motion sensor becomes a position sensor. This functionality can be used for many ranging and imaging tasks—everything from a simple studfinder to an experimental unit able to inspect substructures (beams, pipes, and conduits) buried in a foot of concrete. One of the most hoped-for (and fittingly ironic) imaging applications under study is buried land-mine detection.

As a simple rangefinder, the MIR competes favorably with the widely used ultrasonic units, offering similar range (e.g., 10') and accuracy (e.g., 1"). However, unlike ultrasonics, the MIR is impervious to plastic (so it can be enclosed], ice, snow, and mud.

Therefore, MIR is especially suitable for automotive applications such as backup and blind-spot alarms, active suspension, and the like. The guts of the MIR can also be reconfigured with a laser diode and photo detector, making for a relatively low-cost laser rangefinder.

Another spin includes all sorts of liquid-level detectors and switches since MIRs can see through nonmetallic tanks and are impervious to the gotchas [e.g., condensation, dirt, foam, etc.) that trip up sensors. There's also a design for an electronic dipstick using time domain reflectometry (TDR) to measure everything from oil in your car to acid in a factory (coat the stick in glass) or even flammable materials (the small amount of energy considered intrinsically safe).

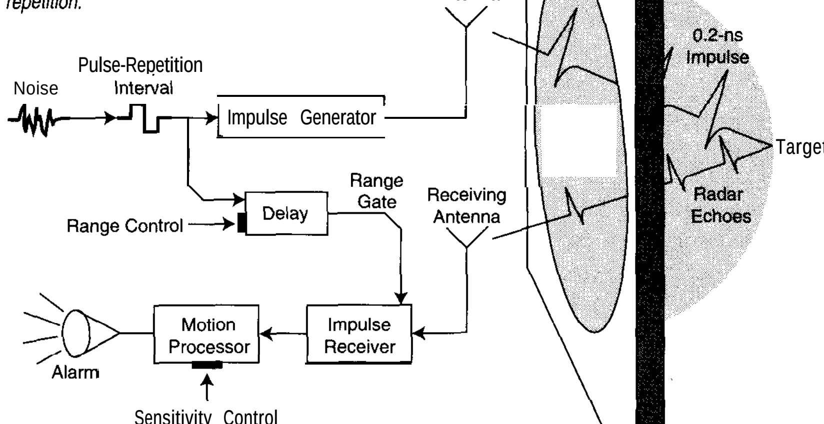
Separating the receiver and transmitter opens up a whole bunch of other applications, too. For instance, a transmitter location in a free space surrounded by receivers can be triangulated-useful for everything from virtual reality to robots.

Also, since the receiver is so low power (few microamps), it can run for a long time on a battery, enabling some interesting installations. Hey, how about an implantable data link so you can download the latest software for your pacemaker?

RADARS R US?

Unfortunately, there no 800 number to call to order an MIR datasheet or

Figure 2—*MIR, like all* radars, *relies on the concept of transmitting* a radio-frequency *pulse* and listening for fhe echo. To *the* basics, *MIR* adds *a range gate and randomized pulse repetition.*



evaluation kit. Don't stop by either. There's no MIR showroom, and barbed wire and armed guards don't make for a pleasant shopping experience.

Instead, the Lab's policy towards selling MIR requires companies to apply for a license in a particular field of use and pay some hefty fees up front ($100k), as well as royalties (3–5%, $25k per year minimum) down the line. Once you're vetted, you can actually get an MIR sample for $5k!

I sympathize with their privatization predicament, but I wonder what the ultimate goal is and whether this scheme best serves it. MIR was paid for by Joe and Jane Taxpayer in the first place. But, you can't get a data-sheet without a Freedom Of Information Act request?

Unless you're willing to ante up, the best you can do is poke around on the MIR Web page and wait for a licensee to make something in your particular field of use-and that seems kind of stifling.

On the other hand, they've successfully completed a number of agreements and are working on more. Certain products, mainly niche fluid-level sensors, are close to production.

At least one licensee reports that making a deal wasn't hard, the tech support (all those Ph.D.s) is excellent, and they would "do it again." It sounds better than dealing with the IRS.

What's the ultimate MIR application? X-ray specs, of course! (And, I bet the Lab could come up with one heck of a "smoke bomb," too.) It all sounds marvy, but I remember that comic-book lesson from long ago.

Fact is, it's going to take time for the promise of MIR, including the bluest-sky applications and the "$10" price tag, to be fulfilled. MIR certainly won't replace all other proven sensor technologies (optic, sonic, or capacitive) in one fell swoop.

Nevertheless, it's a sure bet that the particular advantages of MIR will ultimately enable some rather compelling applications. Whether it's mine detection, a baby monitor, automatic cruise control, or even x-ray specs, it won't take more than a few breakthrough products for MIR to make a big difference. ❑

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.

CONTACT

Michele Lynch
Lawrence Livermore National Laboratory
P.O. Box 808, L-495
Livermore, CA 9455 1
(510) 423-1077
Fax: (510) 422-1796
mir@llnl.gov
http://www-lasers.llnl.gov/lasers/idp/mir/mir.html

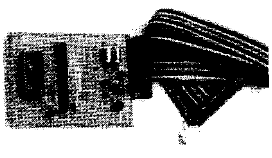
I R S

431 Very Useful
432 Moderately Useful
433 Not Useful

TIRED OF WAITING FOR THE PROMPT ?
Speed up with a ROM DRIVE! Boots DOS and programs instantly. Also used to replace mechanical drive completely for controllers, diskless workstations. The perfect protection from viruses. Easy to install half-size card.

DOS IN ROM!

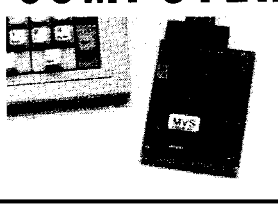
MVDISK1 128k $75
MVDISK2 1 4 4 m 5150
MVDISK3 5 76m $ 1 9 5
$75
Quantity discounts!



$95 EPROM PROGRAMMER
- Super Fast Programming
- Easier to use than others
- Does 2764127080 (8 Meg)

WORLDS SMALLEST PC !!!
ROBOTS ALARMS RECORDERS DOS
THREE EASY STEPS: $27 1K QTY
1. Develop on PC
2. Download to SBC
3. Burn into EPROM $95 SGL QTY
-2 PARALLEL -LCD INTERFACE
-3 SERIAL -KEYBOARD INPUT
-PC TYPE BUS -REAL TIME CLK
-BIOS OPTION -BATTERY OR 5V
FREE SHIPPING IN U.S.
5 YEAR LIMITED WARRANTY

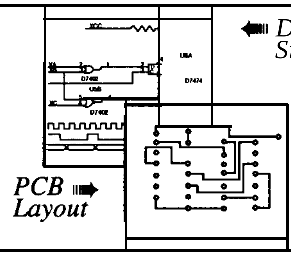
8088 SINGLE BOARD COMPUTER



MVS Box 850 Merrimack, NH (508) 792 9507

#124

Low Cost CAD Software for the IBM PC and Compatibles
Now In Windows™95



- Easy to use schematic entry program (SuperCAD) for circuit diagrams, only $149. Includes netlisting, bill of materials, extensive parts libraries.
- Powerful, event-driven digital simulator (SuperSIM) allows you to check logic circuitry quickly before actually wiring it up. Works directly within the SuperCAD editor from a pulldown menu and displays results in "logic analyzer" display window. Starting at $149 this is the lowest cost simulator on the market. Library parts include TTL, and CMOS devices.
- Analog simulator (mentalSPICE) for $149. Allows AC, DC and transient circuit analysis. Includes models of transistors, discretes, and op amps.
- Circuit board artwork editor and autorouter programs (SuperPCB), starting at $149. Produce high quality artwork directly on dot matrix or laser printers. You can do boards up to 16 layers including surface mount. Includes Gerber and Excellon file output. Autorouter accepts netlists and placement data directly from the SuperCAD schematic editor.
- Low cost combination packages with schematics and PCB design: 2-layer for $399, 16-layer for $649.

Write or call for free demo disks:
MENTAL AUTOMATION, INC.
5415 - 136th Place S.E.
Bellevue, WA 98006
(206) 641-2141 • BBS (206) 641-2846
Internet: http://www.mentala.com

#125

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**

*This month, I'm going to stick with just **two** threads. The first one does a nice job of covering what's necessary to think about when frying **to** protect against various high-voltage and high-current **transients.***

In the other thread, we revisit the low-current relay subject. You have a few more good resources to check out.

ESD

Msg#: *3872*
From: Joseph Lehman To: All Users

I am interested in a discussion on what has been found to be effective ways of providing static protection for external inputs. I have two cases of interest. The first is a 5-V input going to an input pin on a PIC processor. The input on the processor is clamped with internal diodes rated at ±20 mA.

The second case is a differential analog input into an LM358. The differential is done with 1-MΩ resisters in a voltage-divider configuration to ground, and the other side is 1-MΩ voltage-divider resistors to the op-amp output.

Thanks. This place generates some great discussions.

Msg#: 3879
From: Ken Simmons To: Joseph Lehman

Speaking as someone who has to deal with ESD prevention every day, let me have a stab at it.

I'd try using MOVs, transorbs, or similar surge-suppression devices first.

Next, I'd try zener diodes rated slightly higher than the expected device's input level (i.e., a 6-V zener for a 5-V input) or even fast-switching diodes (e.g., 1N4148) clamping the input to V_{CC} and Gnd.

Last, use LC filters in conjunction with MOVs, transorbs, and so forth.

That's all I can think of.

Msg#: 3886
From: Mike Tripoli To: Ken Simmons

Just my two cents, but I believe that the internal diodes on the PIC are all that are needed. Just use a series resistor with the input to the MC to limit the current. This info is in the Microchip databook.

Msg#: 3897
From: Ken Simmons To: Mike Tripoli

That's the bare minimum for ESD or other overvoltage protection. Since I work on military electronics products, what I run into is a little beefier than a simple diode.

To put it another way: would you trust a simple diode to protect your baby from ESD, or would you prefer something more robust?

The extra steps you take to protect your PIC now will pay off in the long run. Remember, 01' Murphy knows many ways of causing problems, especially those which result in your saying "I never thought *that* would happen!" when it does.

Msg#: 3900
From: George Novacek To: Mike Tripoli

> Just my two cents, but I believe that the internal
> diodes on the PIC are all that are needed.

I wouldn't bet my life on it! ESD is known to have a very fast rise time, about l-3 ns, which corresponds to the equivalent EM1 frequency of 300 MHz. Neither diodes nor transorbs are fast enough to clamp it. Because of the finite speed of the clamp, you end up with an extremely narrow pulse which may have 2000 V at 50% humidity and exceed 15,000 V at 5% RH (due to human touch). The microcircuit may not fail right away, but damage will be done and eventually it will die.

As a very minimum, you need a small capacitor in addition to the internal clamping diodes. If you are lucky, parasitic capacitances may do the job, but I would not rely on it. It is always better to have a defined RC constant (some ICs now contain those capacitors on the substrate).

You need to protect a system against essentially four major threats: ESD, EMI, and two indirect lightning effects. Successfully protecting against one does by no means guarantee you are also effectively protecting against the others. At times, the requirements may be contradictory and you have to choose a compromise.

As I already pointed out, ESD represents narrow, high-voltage pulses. You can protect against them with an RC constant (C closer to the solid-state device!) and diode clamps.

CONNECTIME

For EMI, you are generally working with radiated electromagnetic fields in the 10-kHz to 20-GHz range, with field strength of several hundred volts per meter (i.e., this is the voltage a 3' piece of wire would pick up).

Besides good shielding, all leads going into the shielded enclosure must have in-line filters. These are usually "pi," "L," or "T" LC filters, depending on the type of signal, load, and required attenuation.

It might be tempting to assume that the in-line low-pass "pi" filter which starts cutting off at 3 kHz will automatically take care of your ESD problems. But, it won't necessarily. As there are no ideal components out there, filters lose their attenuation at high frequencies due to parasitic impedances.

Lightning strike (single pulse) and transient protection must deal with overvoltage pulses. Select the protection depending on the severity of the transient you want to survive. In many cases, an MOV or a transorb will do the job. MOVs are OK for protecting equipment where a transient surge would be considered unusual. But, they're not a good option otherwise since they deteriorate with each transient they clamp.

For high reliability (and faster speed), you are better off with transorbs. In really abusive environments, you may have to use a two-stage protection with a spark gap (i.e., a discharge tube) at the first stage, where it will clamp at 200400 V, followed by a small resistor and a transorb to maintain the voltage below, say, 5.1 V for logic circuits.

But again, because of the finite conduction speed of the semiconductor devices, a narrow pulse will get through. This must be handled by a filter which is designed for the equivalent EM1 frequency. (If ESD is not a concern, although it usually is, just an LC feedthrough is OK since lightning does not have the ESD fast rise time. But since you need to consider ESD and debouncing, an RC circuit usually follows anyway.)

Last, the second lightning-strike effect you need to protect against is the common-mode burst, which affects all lines and grounds simultaneously. That must be addressed during the circuit design. The major issue here is the functional upset. If the input circuits have a high common-mode voltage range, the clamps can be set fairly high and the circuit operates normally.

A typical strike is a random succession of about twenty l-MHz bursts of different widths, lasting a total of 2 s. If the dynamic range is low, the clamps cut in at a fairly low level (e.g., 5 V). Because of the low-pass filters following the clamps, a DC offset develops on the inputs, which may create either wrong logic input or obliterate an AC signal on analog inputs.

This is just a very brief overview of what you are up against.

Msg#: 3916
From: Jan Verhoeven To: Joseph Lehman

I guess you're acquainted with some quantum mechanics, so you've heard about Heisenberg's equation: ***dp x dv > k,*** in which ***dp*** is the uncertainty about the position of a particle, ***dv*** is the uncertainty about the speed of a particle, and ***k*** is a constant.

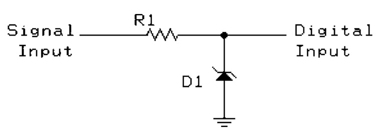
The more precise your knowledge of, say, the speed of the particle, the higher your uncertainty about its position in space.

This relation goes for about any system in which there are several factors at work. It merely says: the more effort you invest in one possible problem, the higher the chance you'll have other problems rocketing sky high.

The solutions you mentioned (clamping diodes and high-resistance voltage divider) are very good and have a low impact on general circuit performance. You could introduce more safeties, but only at an expense. Each and every situation dictates what you can do. I guess there will be no general rule that will apply for all situations.

Nonetheless, I will add my two ***dubbeltjes:***

For making digital inputs foolproof, I mostly use the following circuitry:



R1 is a current limiter, mostly set at 4.7 kΩ or higher (depending on the source), and the zener diode D1 is mostly a 4.7-V or a 4.3-V one.

This way, very high voltages, even with enough power, will not harm my inputs, whereas reverse-polarity signals will simply forward bias the zener and pass through without any problem.

For analog inputs, the resistor divider is a very neat one. If necessary, you could add some power diodes in parallel to them, one going to each power rail. But then the diodes might interfere, since they are not passive components.

For total safety, I would suggest optocouplers. For digital signals, optocouplers are dead easy. For analog signals, they're another piece of cake. But with the newer IL300 type of linear couplers, it can be done. But it'll introduce higher costs all over. And in a consumer product, that can be a real killer.

Msg#: 3975
From: Pellervo Kaskinen To: Joseph Lehman

You already have a definite answer from George Novacek. Here is just a little more to give a different perspective.

The PIC protection appears to be a case of a single-ended circuit, unless you put in an optocoupler. Actually, you might as well. But then, you have to provide the power to the primary of the opto. And the opto device input needs protection against ESD.

You may have been given a little too pessimistic a picture of the reaction time of transorbs (or zener diodes]. Any slowness in the response is predominantly due to the wires, not the silicon!

To really do the protection right, there are four-terminal protective devices that offer next to zero wire length. In some ads I have seen, they indicate values good enough for almost any ESC situation, ***provided you*** can route the signal lines and ground line through the protector. You do not get any improved safety if you run your ground to the IC and just make an extra leg to the protector. No, the idea is that any signal and ground can reach the IC only through the four-terminal clamping device.

I agree with the advice that, after a clamp, there should be a little capacitor. But, the same wire length issues come into picture here. Surface-mount capacitors across the lines between the clamp and the IC are a good way. Otherwise, don't spread the PCB lines. Solder a small ceramic disc capacitor there with minimal wire length, or provide a ground-plane-type multilayer board.

There are limits to how much series resistance you can have at the input of an IC. But for the ESD, you might put in a small choke instead. A 10-H choke with a ¼-W resistor form factor can have a resistance of just a couple of ohms and yet be a virtual open circuit for the 300-MHz front of the ESD pulse.

Now, of course, you might not need protection only against the ESD, as others have pointed out. Then more resistance is desirable. And, there is always the chance that the protective resistors or chokes might not be able to take the 4000+ V of a typical ESD pulse either.

Minimum Relay Currents

Msg#:21634
From: Calvin Krusen To: All Users

I have a question about minimum current ratings for relays.

Are there relays designed specifically for extremely low currents (10 nA to 10 µA) with voltage ratings up to 100 V?

In the past, I've read about relays and switches requiring a minimum current to prevent some sort of buildup on the contacts. I am not too concerned with the resistivity of the relay because I am using a current source (i.e., voltage drop is not a problem).

Msg#:23623
From: Pellervo Kaskinen To: Calvin Krusen

In general terms, most reed capsule relays are quite good in handling the low currents you ask for. They have an inert gas to protect the contacts and thereby do not depend on the signal current to "punch through." Even better are the mercury-wetted types.

Of the open types, only the crossbar-type contacts, and better yet, bifurcated crossbar contact types are reasonably good. Just in case an explanation is needed, bifurcated contacts consist of two halves of the contact, each half able to "make it" because of the flexing of the leaf, even when the other half is already stopping the large motion. The crossbar is like two knife edges at a 90" angle against each other. You get a tiny contact area with plenty of local force. This high pressure pushes any oxide layer away and exposes the underlying metal.

Oh, your relay just has to have gold as the contact material. It can be rhodium plated on top of the gold for a little better longevity. Do not use copper, silver, or silver/ cadmium oxide contacts. They do not last very long. They are used for the higher currents.

As a practical model, an Omron G5V-2-24 relay is a good choice that I'm using. The -24 is for the coil voltage. This relay is in a package like a 16-pin DIP, but taller. It has two form-C contacts, rated 125 VAC or 32 VDC each. You probably want to put in some amount of spark suppression if your load has any inductance, even though your currents are so low.

Msg#:27275
From: George Novacek To: Calvin Krusen

Here we go again! The topic has been discussed on this BBS quite a few times. There are studies available (dating back to the '40s). I posted the titles some time ago, but am too lazy to get up and look for them now. Besides, I doubt you would be able to find them very easily.

The rule of thumb (at least in my industry) is that anything below 10 mA needs gold contacts. Below 100 mA, you have to have at least silver. For very low currents, you want to use "dry" contacts-the better the seal from the atmosphere, the better for the connection.

Finally, if you want to have 1000 V across the open terminals, you have to worry about arcing and simple dielectric breakdown. We used a rule of thumb again: 1000 V can jump over 1 mm of air.

Msg#:23467
From: Lyndon Walker To: Calvin Krusen

Try EAC components. I have samples of the R1A12AH-3 which I ***still*** have yet to take out of the box (another delayed project!). Off the top of my head, I think their mini-

CONNECTIME

mum current spec was around 10 µA (http://www.reedrelays.com/).

Also, look at the Coto-Wabash 7000 and 3600 series. Aromat also lists low-current reed relays in EEM, but the tech support was somewhat less than knowledgeable.

I've found that you generally have to ask the manufacturers. They tend not to publish minimum current specs even when the relays are tested and designed for it.

We invite you to call the Circuit Cellar BBS and exchange messages and files with other Circuit Cellar readers. It is available 24 hours a day and may be reached at (860) 871-1988. Set your modem for 8 data bits, 1 stop bit, no parity, and 300, 1200, 2400, 9600, or 14.4k bps.

ARTICLE SOFTWARE

Software for the articles in this and past issues of Circuit Cellar *INK* may be downloaded from the Circuit Cellar BBS free of charge. It is also available on the Internet at <**http://www.circellar.com/**>. For those with just E-mail access, send a message to info@circellar.com to find out how to request files through E-mail.

For those unable to download files, the software and messages are also available on disk. Software for issues prior to 1995 comes on a 360-KB IBM PC-format disk, one issue per disk. For issues from 1995 on, software comes on a 1.44-MB PC-format disk, with three issues per disk. Disks cost just $12 each. To order Software on Disk, send check or money order to: Circuit Cellar INK, Software On Disk, 4 Park St., Vernon, CT 06066, or use your Visa or Mastercard and call (860) 8752199. Be sure to specify the issue numbers with your order. Please add $3 for shipping outside the U.S.

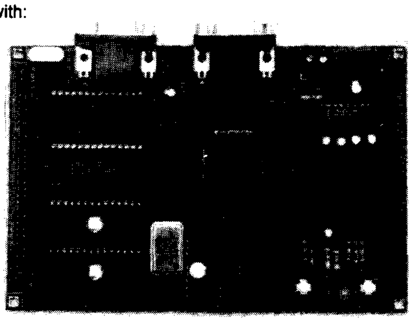
434 Very Useful *435* Moderately Useful 436 Not Useful

CAN-ABLE(TM)

RIGEL Corporation introduces its newest line of boards, the **CAN-ABLEs(TM)**. Our 8 and **16-bit** CAN-ABLE(TM) boards are designed for immediate placement into a Controller Area Network.

The **RMB167-CRI** features the Siemens SAB C167-CR processor, and includes : an RS232 serial port, 111 **I/O** bits, five **16-bit** timers, **20MHz** system clock, built-in CAN interface, 4 channels of PWM, **upto** 1 MB of FLASH **memory** and **upto** 1 MB of SRAM.

THE RIGEL DIFFERENCE
All of our boards come with:
Machine screw sockets
Power on LED
All system signals and I/O available on headers
Sample programs
Circuit diagrams
Assembly language IDE software
READS



Our entire line of boards are programmable in Assembly, "C", and Forth.

FOR MORE INFORMATION VISIT OUR WEB SITE
http://-.rigelcorp.com
or contact us at
RIGEL Corporation,
PO BOX 90040, GAINESVILLE, FL 32607
Telephone (352) 373-4629 FAX(352) 373-1786
sales@rigelcorp.com

#127

4th ANNUAL FIRE FIGHTING HOME ROBOT CONTEST

$1,000 FIRST PRIZE
SUNDAY, APRIL 20, 1997
Trinity College
Hartford, Connecticut

- Robotists and **their creations from all** over will compete in this international event
- Contestants range **from college professors** and rocket scientists to 4th **graders**
- A weekend full **of activities** in**cluding a practice session,** Robot Fair and Robotics Seminars
- Get together **with** interested ***people, trade ideas*** & information, ***and HAVE FUN!***

Come for the weekend or come for **the day,** but **COME** AND SEE THE **FUTURE!**

For more **information and the contest** rules, **contact Jake** Mendelssohn at: **jake.mendelssohn@circellar.com** or 190 Mohegan Dr., **West** Hartford, CT 06117
On the Web: **http://www.trincoll.edu/~robot**

PRIORITY INTERRUPT

When It Costs Nothing, What's It Worth?

One of the more interesting support vehicles for INK has been the BBS. Over the years, thousands of readers have left words of technical wisdom for others. The quality of these postings is superb. Enough so that we often find ourselves apologizing for continual busy signals as hundreds of callers try to squeeze through four incoming phone lines and an Internet link. Expansion to full Internet access is on the schedule.

There is no charge for logging onto our BBS, other than the cost of the telephone call itself. At one time, I looked into adding more phone lines to reduce the bottleneck. In the early days, callers would log on to read or type messages directly. It wasn't unusual for callers to spend 30-45 minutes browsing through the message base. The net result was that maybe only 25-50 callers could get through during peak hours.

As we were about to attempt various remedies, the problem seemed to abate. While advertising might lead you to believe that long-distance phone rates are dirt cheap, apparently that was not reality. When we added the ability to download new messages and upload offline-composed answers as a quick-call connection, virtually all the long-distance people jumped on it. Instead of a half-hour real-time connection, people could transfer the messages in a few seconds. As lines freed up, many more users joined the ranks.

Analyzing costs caused people to establish a superior alternative to anything I might have imposed in the way of connect-time regulation. Short of major system expansion-hard to justify for complimentary services-I could not have wished for a better solution.

As I look at moving the whole database onto the Internet, I see parallels to our previous situation. Unfortunately, the elements affecting performance and throughput are moving in the wrong direction!

Not even a year ago, you could log onto the Internet and dial up a Web site with some assurance that you'd actually connect. Because connect charges were hourly rated, you'd browse a little, download a little, and even enjoy it a little. But, the last time I was on, I spent 20 minutes trying to log onto a half-dozen popular sites. When I finally connected, the real-time screen update was so slow that I went and ate dinner while one download was in progress. What real value is there in rushing to upgrade to the latest 33.4- or 56-kbps modem when your actual throughput is probably 1200 bps?

The real culprit again is cost analysis. I could complain yet go to dinner while still online because I now, like many of you, pay only $19.95 per month for unlimited connect time. For many more users, such as university students or government-supported facilities, access is free. At that cost, why log off at all? At that cost, using the Internet for real-time display of data from a Middle East oil well or the stock ticker on Wall Street appears to be a cost-effective reality.

Unfortunately, perception is not reality. Because Internet access is so inexpensive, young Billy spends 10 hours a day playing interactive video games and you leave your connection on all the time to collect E-mail. Whether the exercise involves simple ASCII text or mega-bandwidth videoconferencing and multimedia, the cost is the same.

Internet access has become too cheap. The result is a system that is becoming worth about as much as it costs. Basic economics dictate that when a commodity is in demand and the price goes down, even more people will buy it. But, when the commodity is finite, only a lucky few can be adequately accommodated.

The Internet has only so much capacity. With access-provider income negatively spiraling (after the positive influx from everyone signing on for $19.95) there's little bonus and even less reward for expanding hardware to keep up with the continuing demand. Let's face the fact that you don't get something for nothing. When something costs nothing, what's it worth?

Steve

steve.ciarcia@circellar.com