EMBEDDED PC SPECIAL BONUS SECTION: EMBEDDED PC

CIRCUIT CELLAR INK®

THE COMPUTER APPLICATIONS JOURNAL

#77 DECEMBER 1996

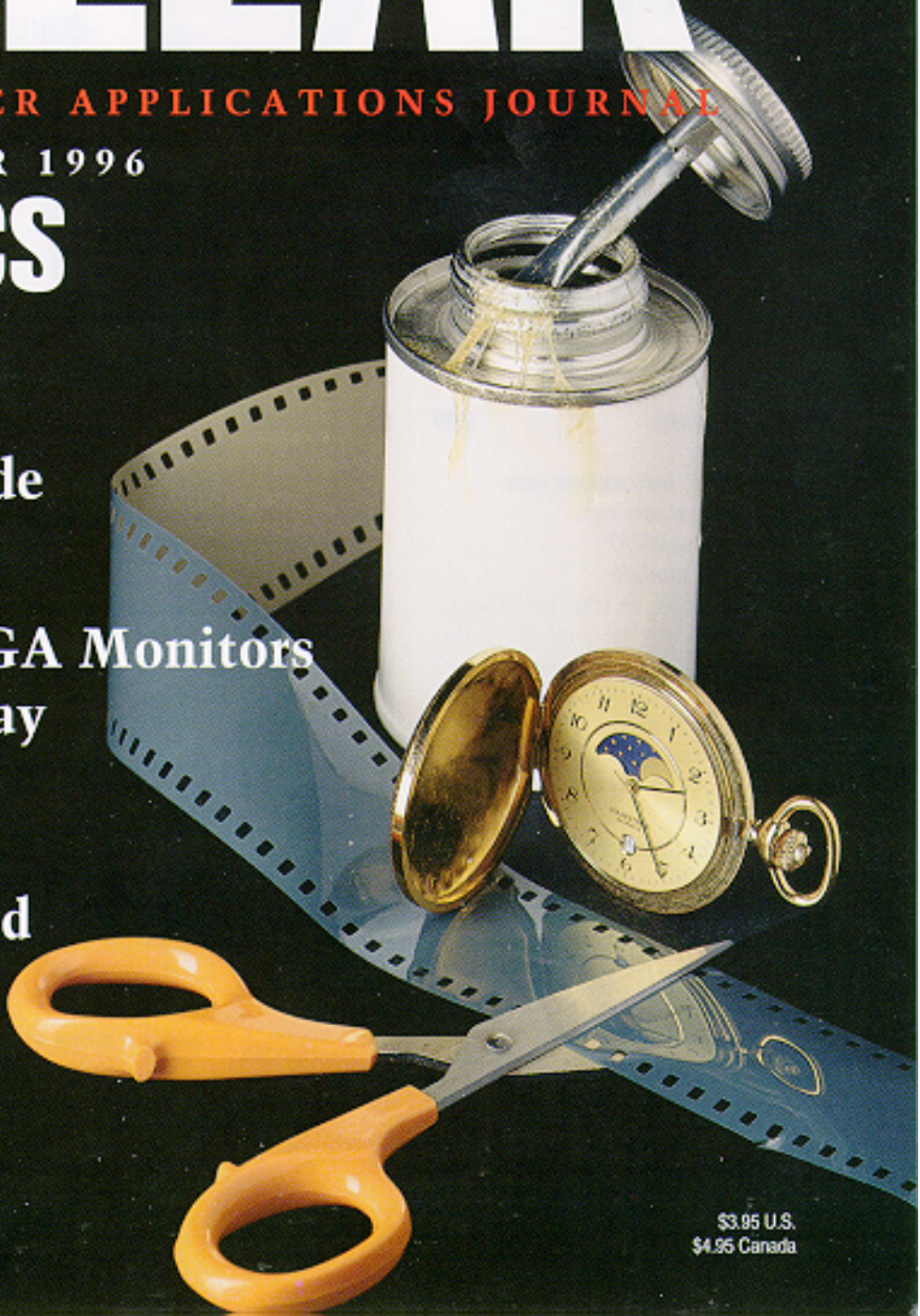
GRAPHICS & VIDEO

Video Timecode Fundamentals

Converting VGA Monitors to Plug-and-Play

GPS Theory

EPC: Embedded Windows NT



12
0 74470 75349 0

$3.95 U.S.
$4.95 Canada

TASK MANAGER

Walt Graphics?

We're always hearing how embedded processors have infiltrated the average person's life. Media gee-whizzes boast that there are so-and-so many 'processors in the average home. Even the Internet is at the forefront, with Web addresses showing up everywhere from TV ads to cereal boxes.

A more subtle trend is the abundant use of computer graphics, especially in television, movies, and print. From scrubbing bubbles to breakdancing infants, graphics can be seen on a daily basis.

There was an editor's note in a recent issue of *Life* in which they admitted to touching up a photo by computer in a previous issue. They did it to remove an element that ruined the composition of the scene. They said such retouching was against their policy, apologized for the lapse, and said it wouldn't happen again.

How many people realize that the scenery for the ballroom scene in *Beauty* and the *Beast* was computer generated, with the hand-drawn characters placed on top? The animators admitted that the scene wouldn't have been possible using traditional animation techniques.

In The *Hunchback of Notre Dame,* the confetti in the town square during the Feast of Fools is computer generated. In the same movie (and, to a lesser extent, in The Lion *King),* the fuzzy focus on scenery either very close to or very far from the "camera" was done with computer processing.

But, of course, the ultimate example is Toy *Story.* Done completely on computer, it shows just how close we are to the day when traditional animation is going to be completely replaced by computer graphics.

Speaking of videos, in our first feature this month, Ingo Cyliax deals with video timecodes-how they're done by industry and how you can use them to edit your own creations.

On the display side of things, Plug-and-Play is all the rage in the PC world. Peter Sorrells and Shannon Poulin show what little work is required for manufacturers to modify existing monitor designs to add PnP capability.

Next, Bruce Hubbard covers the fundamentals of image filtering. It's amazing what can be pulled from an image when it's processed digitally.

In our final feature, Do-While Jones homes in on the Global Positioning System (GPS) and provides the basics for how and why it works. We're well-positioned for next month's even more in-depth look.

In our columns, yours truly has finally put the author hat back on to kick off our new MicroSeries column with a look at some sample XPRESS code for the HCS II. Jeff continues his discussion of design tools with an overview of PC board layout basics. And, Tom warms up to Motorola's new ColdFire processor family.

In Embedded PC, Kent Tabor and Clint Hanson explore a new standard for passive backplanes that tries to address the past, present, and future in one package. Next, Naren Nachiappan finishes up his look at real-time operating systems by checking out what it takes to embed Windows NT.

In PC/104 Quarter, Ed Foster describes some techniques for designing switching power supplies that withstand the rigors of mobile environments, and in Applied PCs, Fred talks about Phar Lap's TNT Embedded ToolSuite.

editor@circellar.com

CIRCUIT CELLAR®

THE COMPUTER APPLICATIONS JOURNAL

FOUNDER/EDITORIAL DIRECTOR
Steve Ciarcia

PUBLISHER
Daniel Rodrigues

EDITOR-IN-CHIEF
Ken Davidson

PUBLISHER'S ASSISTANT
Sue Hodge

MANAGING EDITOR
Janice Marinelli

CIRCULATION MANAGER
Rose Mansella

TECHNICAL EDITOR
Elizabeth Laurençot

CIRCULATION ASSISTANT
Barbara Maleski

ENGINEERING STAFF
Jeff Bachiochi

CIRCULATION CONSULTANT
John S. Treworgy

WEST COAST EDITOR
Tom Cantrell

BUSINESS MANAGER
Jeannette Walters

CONTRIBUTING EDITORS
Rick Lehrbaum
Fred Eady

ADVERTISING COORDINATOR
Dan Gorsky

NEW PRODUCTS EDITOR
Harv Weiner

ART DIRECTOR
Lisa Ferry

PRODUCTION STAFF
John Gorsky
James Soussounis

CONTRIBUTORS:
Jon Elson
Tim McDonough
Frank Kuechmann
Pellervo Kaskinen

CIRCUIT CELLAR INK®, THE COMPUTER APPLICATIONS JOURNAL (ISSN 0896.8985) is published monthly by Circuit Cellar Incorporated, 4 Park Street. Suite 20, Vernon, CT 06066 (860) 875-2751. Periodical rates paid at Vernon. CT and additional offices. One-year (12 issues) subscription rate U.S.A. and possessions $21.95, Canada/Mexico $31.95. all other countries $49.95. All subscription orders payable in US. funds only, via international postal money order or check drawn on U.S. bank.

Direct subscription orders and subscription related questions to Circuit Cellar INK Subscriptions, P.O. **Box 698, Holmes, PA 19043-9613 or call (800) 269-6301.** POSTMASTER, Please send address changes to Circuit Cellar INK, Circulation Dept., P.O. Box 698, Holmes, PA 19043.9613.

Cover photography by Barbara Swenson
PRINTED IN THE UNITED STATES

Circuit Cellar BBS—24 Hrs 300/1200/2400/9600/14.4k bps.8 bits, noparity, 1 stop bit, (860) 871-1988; 2400/9600 bps Courier HST. (880) 871-0549. World Wide Web: http://www circellar com/

EMBEDDED PC

See Pages 51–80 for our Special Bonus Section

INSIDE ISSUE 77

READER I/O

$500 WEB BROWSER FOLLOW-UP

As the author of the $500 Internet Web Browser article (*INK* 73), it's been great to hear directly from *INK* readers. Thanks to those who took the time to write.

The feedback I found most interesting was what some had concluded in their own work-the modem link rather than the CPU is the performance bottleneck. In fact, the first fax I received was from an Australian. That really drove home the international scope of the topic (and your magazine!).

I would also like to thank the individuals who really did the development work on this project-Hamilton Hallmark's John Bentham, Mike Givens, and John Butler and QNX's Larry Parks and Phil Oaks. My contribution as the author pales in comparison to the work they did in developing and configuring the system. Thanks again!

Brad Reed
brad.reed@ radisys.com

BETTER ISN'T ALWAYS BIGGER

While reading Steve's "Software-The Real Generation Gap" (*INK* 74), I found myself nodding in agreement about an issue that has peeved me for years-the increasing size and decreasing performance of most PC software.

However, while I strongly agree with how Steve applies this to today's desktop OSs, I don't think that a target system running under a windowed environment must have the same (read "expensive") hardware attributes as its development system.

Properly designed, a PC OS and its windowing system don't have to eat gobs of memory and processor cycles to provide rich functionality and responsive performance. Case in point-the QNX real-time POSIX OS and its windowing system, the Photon microGUI. Because both QNX and Photon are based on a modular microkernel architecture, both can be scaled up to provide a full-blown development environment ***or*** scaled down for a very small resource-constrained embedded system.

You can code, test, and debug a "windowed" embedded application on a fully loaded Pentium and run that same application-with a scaled-back version of the same OS and windowing system-on, say, a '386EX-based system with very little memory and no hard disk.

Steve, the computer revolution you asked for is still happening on the software side!

Paul N. Leroux
paull@qnx.com

THAT LITTLE SOMETHING EXTRA

Since I started reading ***Circuit Cellar INK,*** I have a better feel of what's up in the silicon market. I receive more information from ***INK*** than from original company databooks. The databooks are good, but the knowledge shared by ***INK*** authors adds that little bit extra to help me fully understand what the chips are all about.

Specialist designers have often wondered why I had insight into issues only they'd know about. My prime source for such information has always been ***INK.***

Jan Verhoeven
The Netherlands

NO MORE LONG-HAIRED GEEKS

The trend toward common platforms for PC use in SBC embedded systems breaks the myth that you have to be a long-haired wire geek to understand and build SBC advanced controllers.

Clearly, I'm a PC programming guy. I'd like to learn significantly more about embedded systems, particularly SBC running small kernels.

Bill **Andreas**
bill_andreas@cscmall.csc.com

Contacting Circuit Cellar

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

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

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

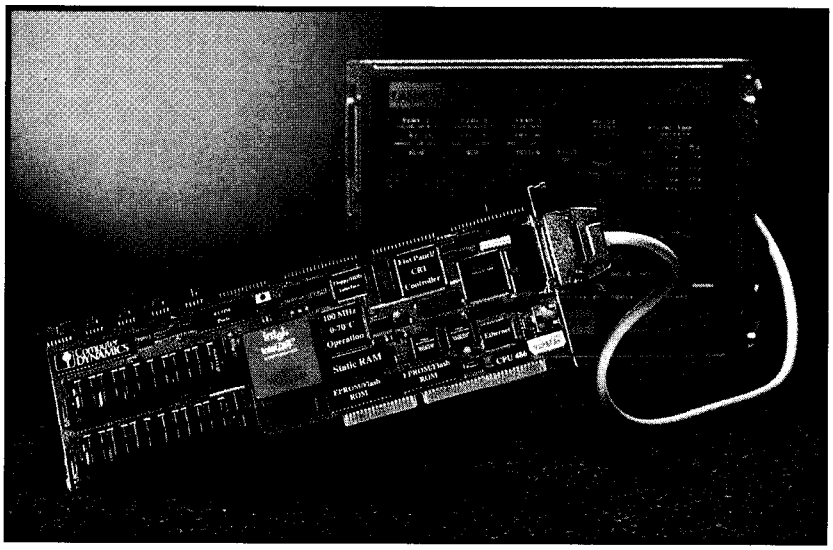
INDUSTRIAL SBC

The CPU-486 provides '486DX4 CPU speeds up to 100 MHz and features two ways to directly interface to flat-panel displays. It's the first ISA-bus board to offer a Low-Voltage Differential Signaling (LVDS) flat-panel interface. LVDS lowers EMI/RFI, enables video signals to travel 30' without degradation, and multiplexes color data signals so that fewer conductors are required in the video cable.

In the first method of interfacing to a flat panel, a 50-pin SCSI connector carries the LVDS signal to the panel and returns RS-232 data to the onboard COM port. This technique is useful for flat-panel display systems using a touch screen. In the second method, the CPU-486 has a header connector which makes the flat-panel interface available in TTL format instead of LVDS.

The CPU-486 includes two right-angle (low height) SIMM sockets for up to 64 MB of DRAM, 2 MB of flash ROM optional, two EPROM sockets, and one SRAM socket for up to 512 KB of battery-backed SRAM. Software support is also provided so application programmers can remotely download programs to flash ROM through a COM port.

The board offers four COM ports (two can be RS-485), an Ethernet 10BaseT interface, an IDE hard-disk interface, floppy and printer port interfaces, a keyboard, and speaker ports.



The CPU-486 is a full-size, low-profile PC-bus board measuring 13" x 4.2" and, when using the TI486DX2 CPU without a fan, it's only 0.5" high (1" with the 'DX4 CPU). The board consumes 11 W and is rated for 0-70°C. Mounting holes and an onboard auxiliary power connector allow busless, stand-alone operation.

The CPU-486 is priced at $1995.

Computer Dynamics
7640 Pelham Rd. • Greenville, SC 29615
(864) 627-8800 • Fax: (864) 675-0106
sales@cdynamics.com

#500

MINIATURE COLOR CAMERA

The **ColorSentry** CCD-835C is a miniature color CCD camera about the size of a pack of cards. It features a ¼" sensor and built-in electronic shutter that eliminates the need for an autoiris lens. It has a resolution of 325 lines and a sensitivity of 0.5 lux. Combined with a digital autowhite balance and backlight compensation, the unit offers extremely accurate color reproduction.

The CCD-835C comes complete with a 4-mm lens, 12-VDC power module, and G-20 universal mounting. An optional 2.5-mm wide-angle lens and 6.5, 8-, and 12-mm lenses are available.

The CCD-835C has a suggested retail price of $413.

CCTV Corp.
280 Huyler St. • S. Hackensack, NJ 07606
(201) 489-9595 • Fax: (201) 489-0111

#501

NEW PRODUCT NEWS

LOW-COST VIDEO OP-AMPS

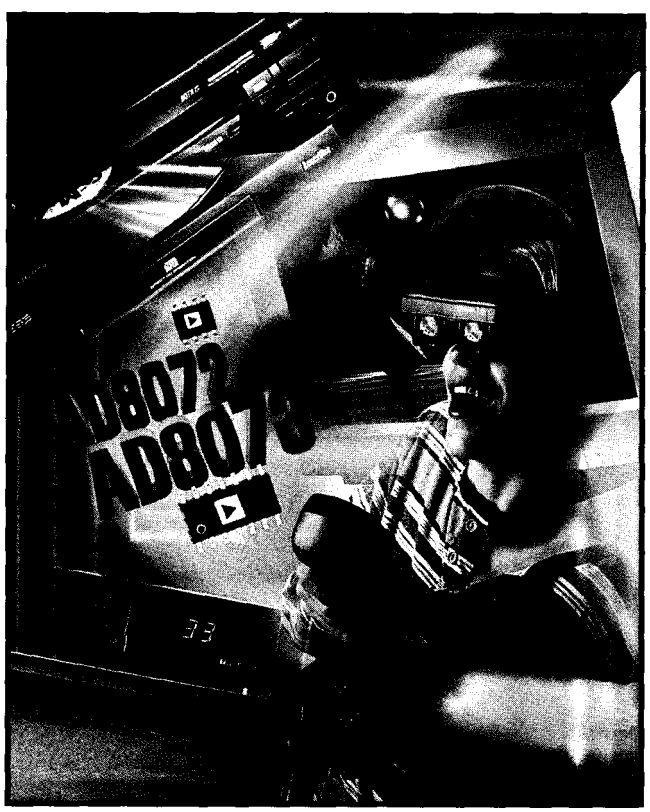
The AD8072 (dual] and AD8073 (triple) high-speed amplifiers exhibit solid video-signal performance with 0.05% differential gain and 0. 1° differential phase error. Available in 8- and 14-pin narrow-body SOIC packages, respectively, they replace equivalent video-amplifier circuits consisting of up to 25 discrete components. For fast, general-purpose applications, the chips combine 100 MHz of bandwidth (-3 dB) with 0.1-dB gain flatness to 10 MHz, a 500-V/µs slew rate, and 20-ns settling time to within 0.1%.

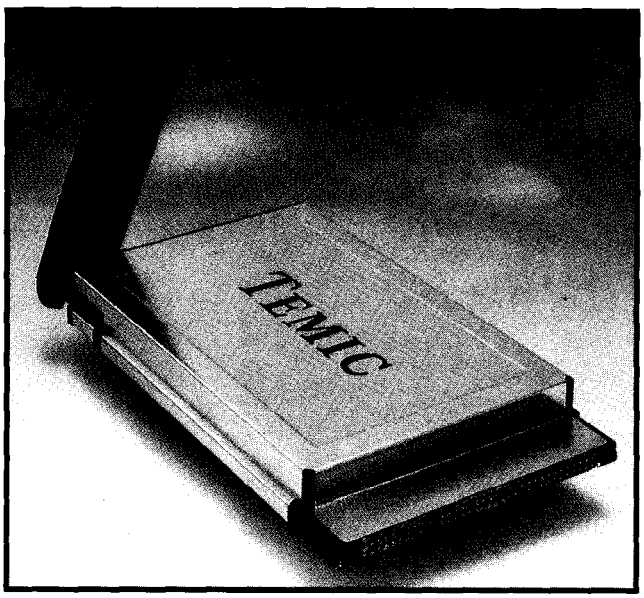
For low-power designs ranging from portable consumer goods to business video applications, both current feedback amps operate from either ±5 V or a single +5-V supply and guarantee less than 5 mA of current per channel. They are optimized to drive 150-R loads for standard RGB, YUV, and S-video signals, and they offer a single-chip solution. Applications include computer VGA plug-in cards, MPEG video systems, and video games.

In 1000-piece quantities, the AD8072 and AD8073 are priced at $1.75 and $2.25, respectively.

Analog Devices, Inc.
P.O. Box 9106
Norwood, MA 02062-9106
(617) 937-I 428
Fax: (617) 821-4273
http://www.analog.com/

#502





VERSATILE PC CARD

Temic presents a PCMCIA Type III PC Card that combines an analog AMPS cellular- and landline-compatible phone with a fax/data modem capable of transmission speeds up to 14.4 kbps. Weighing less than 3 oz., the APM391 comes equipped with a miniaturized, removable antenna and three accessory connectors.

An earphone/microphone jack offers hands-free operation when used with a combination earpiece speaker and bone-conductive microphone. A 15-pin Honda connector allows conversion to landline DAA (Data Access Arrangement] operation. A DAA consisting of a dongle and telephone cord is optional. An antenna connector lets you replace the included %-wave monopole-type antenna with an external antenna.

The APM391 provides four default operation modes—AMPS cellular modem, AMPS voice, PSTN voice, and PSTN fax/data modem. The APM391 switches automatically between these options depending on whether the earphone jack or landline connector are working separately or together. Data, fax, VoiceView, and answering-machine modes can be manually selected.

With Carrier A or B capability, the APM39 1 is capable of extended range, 832-channel operation with 600 mW of transmit power. It requires a minimum 650-mA power supply. The unit operates with standard modem, fax, and voice software applications supporting a standard AT command set.

The APM391 is priced at $400.

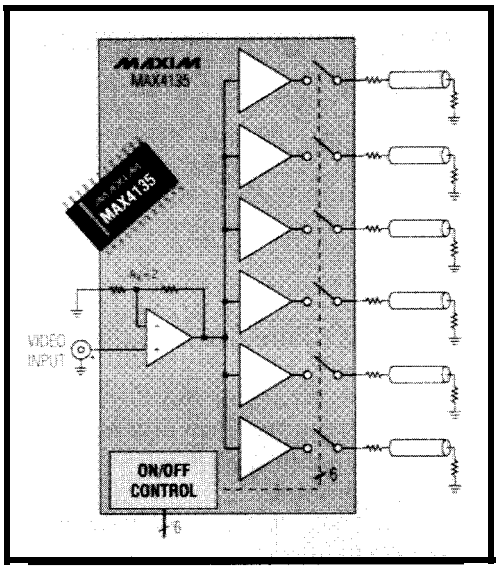
Temic Microsystems
P.O. Box 54951 • Santa Clara, CA 95056-0951
(408) 567-8220 • Fax: (408) 567-8995 **#503**

NEW PRODUCT NEWS

VIDEO DISTRIBUTION AMPS

The **MAX4135–MAX4138** video distribution amplifiers combine high speed with fast video switching. Typical applications include high-resolution RBG monitors, high-speed analog bus drivers, RF signal processing, composite-video preamplifiers, and video switching and distribution.

All four products include an input amplifier and an independently controlled unity-gain buffer for each output. Onboard control logic lets you select any combination of the different signal outputs. The MAX4135 and MAX4136 have one input and six outputs. The MAX4137 and MAX4138 have one input and four outputs.



Each device features a slew rate of 1000 V/us, gain flatness of 0.1 dB to 40 MHz, output-current capability of 70 mA, and low differential gain/phase errors. Fast channel switching (25 ns) enables rapid video multiplexing in applications that display a picture within a picture.

The MAX4135–MAX4138 amplifiers come in 24-pin, wide SO packages with an extended industrial temperature range of -40°C to +85°C.

The MAX4135 and MAX4136 are priced at $5.90 and the MAX4137 and MAX4138 cost $4.50 in 1000-piece quantities.

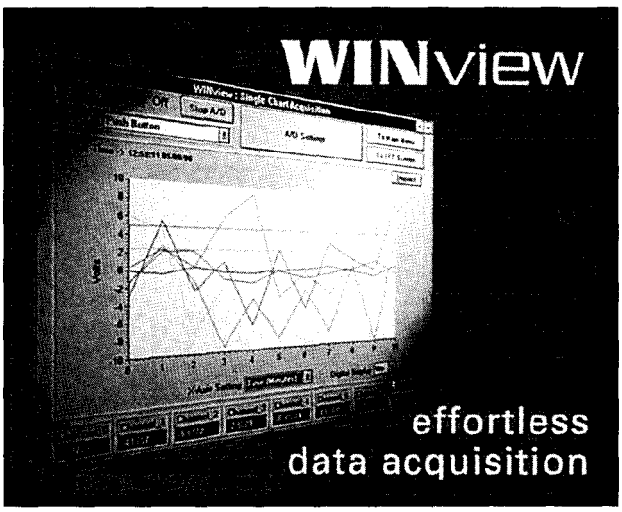
Maxim Integrated Products
120 San Gabriel Dr.
Sunnyvale, CA 94086
(408) 737-7600
Fax: (408) 737-7194
http://www.maxim-ic.com/

#504

105

NEW PRODUCT NEWS

DATA-ACQUISITION SOFTWARE



WINview is a low-cost, "no learning curve" data-acquisition software package for Windows 95 and Windows 3.11. WINview's intuitive buttons, pull-down menus, and radio-button selections enable users to acquire, display, and store data from data-acquisition boards. This high-speed package acquires data at up to the top speed of the data-acquisition board or as slowly as once per month.

The software displays collected data in units of volts, degrees, microstrain, or any other engineering unit. In addition, inputs can be graphed in seconds, minutes, hours, days, samples, or in relation to another input (x vs. y). WINview has advanced disk-logging options that include saving data as ASCII text, ASCII text with time/date stamping, and raw binary for very high-speed and efficient disk logging. It also displays stored data files.

WINview displays inputs as a single graph or separate graphs per channel. It has selectable x-axis settings, datastreaming to disk at up to 333 kHz, and options for 10 different time/date stamps while logging to disk. WINview can apply any math formula to any channel and review stored waveforms.

WINview sells for $99. A free copy of the **WINview demonstration disk** can be downloaded from the company's Web and ftp sites.

ADAC
70 Tower Office Pk.
Woburn, MA 01801
(617) 935-3200
Fax: (617) 938-6553
http://www.adac.com/
ftp://ftp.adac.com/pub/adac/

#505

8-CHANNEL REMOTE ADC/CONTROLLER

The Model RAD128 is a low-cost, eight-channel, A/D converter/controller that provides an optically isolated serial interface to any host computer with an RS-485 port. Typical applications include remote data acquisition, process monitoring and control, supervisory control, energy management, building automation, security system, and other remote applications.

The Model RAD128 is housed in a rugged, stackable, NEMA4 enclosure. It functions individually or with up to eight 16-channel AIM-16P analog input multiplexer/conditioners. In this case, up to 128 single-ended or differential analog inputs can be accepted. Each of the RAD128's eight digital I/O lines may be programmed as either input or output. There are no switches to set because all programming is accomplished in software.

The Model RAD128 also features eight single-ended input ranges (±10, ±5, O-10, or O-5 V), an onboard crystal clock with three counter/timers for precision-timed A/D conversion, and a watchdog circuit to provide automatic reset on startup or microprocessor failure. ASCII-based software allows the unit to be used with virtually any host computer.

The Model RAD 128 sells for $385.

ACCES I/O Products
9400 Activity Rd.
San Diego, CA 92126
(619) 693-9005
Fax: (619) 578-9711

#506

FEATURE ARTICLE

Ingo Cyliax

Video Timecode Fundamentals

Want to condense your video collection into a few tapes of your favorite scenes? Ingo can help. After reviewing timecodes, he shows us how to implement a simple timecode reader for consumer-grade VCRs.

If you're like me, you have rows and rows of home videos. But, each video—while chronological-is a jumble of scenes that doesn't tell a story. Eventually, I want to edit the most memorable scenes into a few tapes.

There are several ways of doing this. One involves pressing the pause button on a recording VCR while playing back a video on another recorder. But, there is a better way.

The professional world uses edit controllers, but they can be expensive. So, I decided to build an edit controller on one of my workstations.

But, first things first. Before I can build an edit controller, I have to learn about timecodes.

In this article, I'll describe video timecodes and how they are used. I'll also look at a simple timecode reader as a building block for my edit controller. The Vertical Interval Time Code (VITC) reader I describe extracts these timecodes from commercial video tapes and network broadcasts.

TIMECODES

Timecodes are essentially "real time" labels. They have their roots in the missile-testing industry of the '50s. A method was needed to label flight data acquired during test flights and

recorded on multitrack analog data recorders.

The labeling was done by encoding a representation of the real time onto one track of the data recorders. The time information on the tape was called tape time.

The collected data could be analyzed, usually at a slower speed, when played back and correlated with data from other tapes. These timecodes also synchronized test equipment at multiple locations to a single time standard.

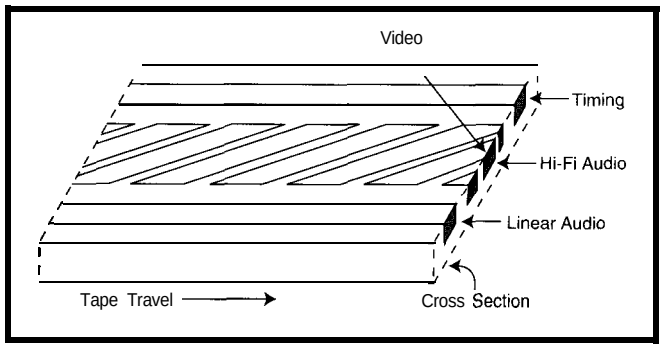


Figure 1—*AVHS has two types of tracks. The audio and timing tracks are recorded longitudinally. Video is recorded diagonally across the tape with a rotating head. Hi-fi audio is carried along with the video tracks.*

In the '60s, videotape recording technology matured to the point that it could be used to edit raw footage into a program. During this time, various timecode schemes were used (many based on traditional time codes) until the Society of Motion Picture and Tone Engineers (SMPTE) standardized their use. These standardized timecodes were also adopted by the European Broadcasting Union (EBU) and are thus called SMPTE/EBU timecodes.

Some professional videotape recorders (VTRs) and editing systems use proprietary timecodes internally. However, they're converted to SMPTE timecodes when making copies for noncompatible equipment.

One notable exception is the Sony RC timecode, which is showing up in so-called pro-sumer equipment (e.g., camcorders and Sony VCRs). Sony RC timecodes are sent via the remote-control channel (Lan-C).

The most obvious use of video timecodes is for editing visual and audio tapes. The original source material is usually recorded via expensive professional video and audio equipment that has timecodes already on it.

Typically, a copy of the original master tapes (including the timecodes) is made in a less expensive format (e.g., VHS tape). The video is then used to preview and edit the film before it is finished processing.

These edit dubs are used to select scenes for the final product. This log, called the edit decision list (EDL), used to be done on paper but is now done with computers.

The EDL is like a computer program that lists the start and end times of each section and notes which effects [e.g., fade, cut, etc.) are used to switch between the sections. The times on the EDL are derived from the timecodes on the edit dubs, which are the same as on the master.

Once the EDL is complete, it and the master materials are sent to a post-production studio (with expensive equipment) where editors (with computers) produce the final tape.

Typically, postproduction studios charge by the hour and are very expensive, which is why producers try to reduce the amount of time editing the tapes in the studio and automate this process as much as possible. Many times a coarse-edit tape is made from the dubs so the program can be reviewed before the fine editing of the studio.

Sound recordings are usually made with separate audiotape recording equipment. During professional shoots, several microphones record on separate audio tracks. These are mixed later when special effects and music are added. The soundtrack is synchronized to the visual information on video or film by recording Longitudinal Time Codes (LTC) on a separate track on multichannel audio recorders.

SMTPE timecodes also contain user bits which are, as the name implies, bits which can be used by the user (e.g., scene annotations). These allow a program to read a tape and extract labeling information such as:

0:00:01 Scene 3, take 1
0:01:23 Scene 3, take 2...

With the user bits, you can record other data. Consider a coach who videotapes an athlete with their physiological data. Heart and breathing rates are recorded from a monitor on the athlete and transmitted via an RF link to the VTR. Using a timecode reader, the coach could correlate that data with the action on the tape.

INDUSTRY TIMECODES

Before I talk about how VITC and LTC are formatted, let's see how VHS videotape is put together. To maximize the recording bandwidth (high-speed tape movement) and minimize the amount of tape used (low-speed tape), video signals are encoded with a rotating head. It records diagonally across the tape as the tape moves.

In VHS, audio is recorded conventionally by stationary heads at one edge of the tape. This type of audio track is called a linear or longitudinal track since it runs along the edge of the tape. On the other edge of the tape

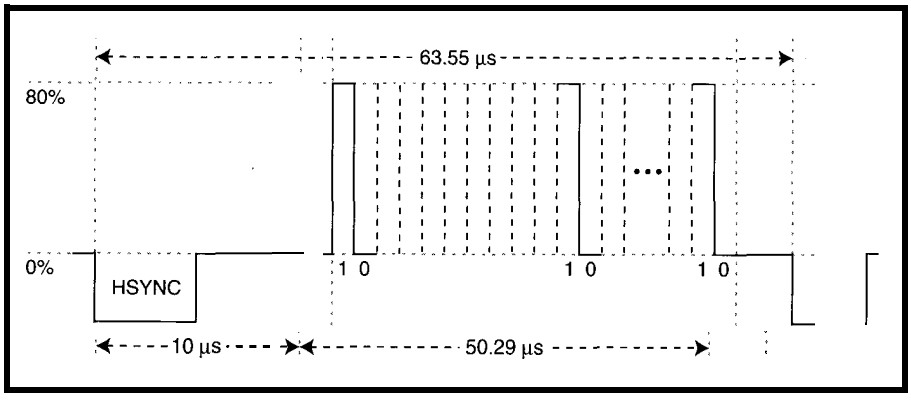


Figure 2—*The VITC signal is encoded to look just like a horizontal line of video. After the horizontal sync pulse, each bit is encoded by representing a 1 as 80% white and a 0 as 0% white (black).*

is a synchronization track which coordinates the forward motion of the tape with the rotating head via a servo loop.

The original linear audio tracks on VHS tape are very narrow and have a single sound channel. In linear stereo mode, this track is divided in two, each having less frequency response.

Hi-fi stereo is recorded by modulating a high-frequency audio carrier and recording it along with the video signal using the rotating head. Since the audio subcarrier is at a much lower frequency than the video carrier, it can be recorded deeper in the tape substrate than the video signal (see Figure 1).

The SMPTE/EBU defines both the LTC and VITC formats. LTC is recorded on linear audio or special timecode tracks longitudinal to the tape edge. VITC can only be used for NTSC/PAL video signals since it is encoded in one of the horizontal lines during the vertical blanking interval (VBI).

VITCs are encoded in the video portion of the VBI. This method is similar to how closed-caption codes are enscripted ("Exploring the Vertical Blanking Interval," *INK 45).*

The most common location is at lines 7, 11, 270, and 274. The 90 bits of information including the sync and data bits are encoded in NRZ (low-0 and high- 1). By making the amplitude of the signal approximately 0–80% of video white levels and using a data rate of 1.789 Mbps (i.e., half the colorburst frequency), the signal can be processed with standard video-processing elements and no signal degradation.

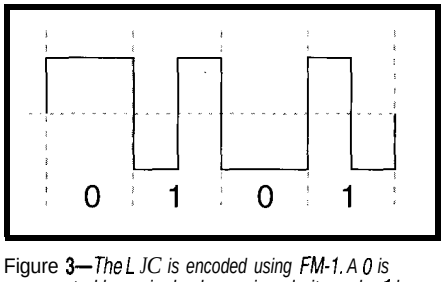


Figure 3—*The L JC is encoded using FM-1. A 0 is represented by a single change in polarity and a 1 by two changes in polarity within a bit cell. FM is said to be self-clocking since the bit clock can be recovered from an FM-encoded datastream by locking a PLLto the signal.*

Bit Position	Description	Bit Position	Description
0	sync 1	42-45	minutes
	sync 0	46-49	user group 5
2-5	frames	50	sync 1
6-9	user group 1	51	sync 0
10	sync 1	52-54	tens of minutes
11	sync 0	55	binary group flag
12–13	tens of frames	56–59	user group 6
14	drop frame flag	60	sync 1
15	color frame flag	61	sync 0
16–19	user group 2	62-65	hours
20	sync 1	66-69	user group 7
21	sync 0	70	sync 1
22-25	seconds	71	sync 0
26–29	user group 3	72-73	tens of hours
30	sync 1	74	unassigned
31	sync 0	75	binary group flag
32-34	tens of seconds	76-79	user group 8
35	field mark	80	sync 1
36–39	user group 4	81	sync 0
40	sync 1	82-89	crc (x^8+1) for O-81
41	sync 0		

Table 1—*VITC* frames are *encoded as video-/eve/ signals in one of the horizontal lines* during the vertical blanking interval. *They* usually *occur on fines* 7, 11, 270, or 274.

A VITC line is shown in Figure 2. Table 1 shows the functions of the 90 data bits.

VITC uses an eight-bit CRC checksum to detect data corruption [e.g., dropouts). This makes it easy to build a simple VITC decoder as I'll show you later. The polynomial used is (x^8 + 1).

LTC is encoded using FM- 1. A single polarity change represents 0, and two transitions represent 1. Since FM- 1 channel coding is self-clocking, it's easy to recover the bit clock when the tape is played back at a higher or lower speed than normal. Figure 3 shows the signal levels and channel encoding, and Table 2 shows the bit assignments.

LTC doesn't have a checksum, but the check and equalization bits can be used for error checking and framing. LTC uses asymmetric framing bits, so it's easy to detect tape direction.

You're probably wondering why two timecode formats are used with videotapes. There are several advantages of using both kinds of timecodes due to how they're encoded.

VITC rides along with the video signal, so no extra tracks are required. Any equipment that processes video signals passes VITC without alteration. Also, since it is part of the video signal, each frame of video can be uniquely addressed.

The downside is that since it's part of the video signal, it can't be changed without rerecording the signal, which degrades video quality.

LTC, on the other hand, can be recorded by simply dubbing the audio track. However, LTC signals have problems at high shuttle speed since the LTC signal requires high-bandwidth audio channels and heads for playback. Also, since they are recorded on linear tracks, they can't be played back while the tape is running extra slow or paused.

Timecode readers in professional video equipment usually read both timecodes and select the most reliable timecode signal for specific modes of VTR operation (e.g., pause, play, and high- and low-speed shuttle).

SMPTE timecodes are identified by their frame rate. A 24-frame timecode (i.e., the frame number goes from 0 to 23) is denoted as SMPTE-24 (film). There is also SMPTE-25 (PAL) and SMPTE-30 (NTSC). Film and PAL timecodes have an integral number of frames per seconds.

However, NTSC has a slight problem. It is broadcast on a 59.97.Hz field rate. So, there are slightly less than 30 frames in each second.

You can deal with this by counting the number of frames normally (O-29), but this introduces timing errors when synchronized with nondrop sources. You can also drop a frame every time the timing error approaches one frame. A bit in the timecode (drop frame) indicates how this is handled when the timecode is calculated.

OTHER TIMECODES

Timecodes used for data logging and synchronization were standardized by the Inter-Range Instrumentation Group (IRIG). The most popular version, IRIG-B, is still used to synchronize equipment to time and frequency references such as GPS time receivers. A modified version of IRIG-H is used by NIST to broadcast standard time signals via radio (WWV/WWVB).

There are other timecodes, like NASA's 36-bit timecode, XR3, and 2137. But, since they're similar to IRIG timecodes, I won't describe them here.

IRIG-{A,B,G}, a hierarchy of timecodes, are pulse-coded AM signals or tone bursts. The carrier tones vary-1 kHz for IRIG-B, 10 kHz for IRIG-A, and 100 kHz for IRIG-G.

Data is encoded by counting cycles. A 0 is represented by 2 cycles, a 1 by 5 cycles, and a position or framing mark by 8 cycles of carrier.

Bit Position	Description	Bit Position	Description
o-3	frames	36–39	user group 5
4-7	user group 1	40–42	tens of minutes
8-9	tens of frames	43	binary group flag
10	drop frame flag	44–47	user group 6
11	color frame flag	48-51	hours
12-15	user group 2	52-55	user group 7
16-19	seconds	56-57	tens of hours
20-23	user group 3	58	unassigned "0"
24-26	tens of seconds	59	binary group
27	phase correction (even number of zeros)	59	phase correction
		60–63	user group 8
28-31	user group 4	64-79	binary sync "0011111111111101"
32-35	minutes		

Table 2—*LTC* contains almost *the same information as VITC.* Nonsymmetric *synchronization bits enable LTC readers to extract the data reliably at various tape speeds and in any direction.*

IRIG-{A,B,G} use the same bit assignments as in Table 3. An IRIG frame is a sequence of 10 subframes of 10 bits each. Each subframe is delimited by a framing mark (8 cycles), so there are 100 bits per frame.

Since pulse length is related to the carrier frequency, the repetition rate is also dependent on it. IRIG-B frames repeat every 1 s, and IRIG-A and IRIG-G repeat 10 and 100 times per second.

IRIG-B timecodes with their 1-kHz tones are easily used with audio equipment (e.g., tape recorders and audio tracks on videotapes).

IRIG-H is a little more specialized than IRIG-{A,B,G}. It uses a 100-Hz carrier but is only 60 bits long, so it transmits within 60 s. By the same principle, two, five, and eight cycles encode the 0, 1, and framing bits.

WWV/WWVH uses a modified version of IRIG-H to transmit UTC over the radio. Table 4 shows IRIG-H's bit assignments as sent by WWV/WWVB.

The Navstar satellites used for GPS contain their own atomic real-time clocks. Each satellite transmits its current time via the Coarse Acquisition Code (C/A Code).

It also transmits data about its location, corrections needed to synchronize the satellite to GPS time, and the offset needed to calculate UTC time (see Do-While Jones' series on GPS, ***INK 77*** and 78).

Even though the timing information in the C/A carrier is purposely skewed due to selective availability, you can achieve accuracies of better than 1 us. Better precision (<300 ns) is possible if a disciplined oscillator is used at the receiver and the time signal from the GPS satellites is averaged over a long period.

SIMPLE **TIMECODE** READER

Let's look at a simple VITC reader. VITC is used by networks so TV stations that rebroadcast network programming can cue local advertising at appropriate times. Some commercial tapes also have VITCs on them so no modifications are needed by consumer-grade VCRs.

Video sources typically have levels of 1 V_{p-p} that range between 0.5 and 2 V_{p-p}. VITC uses 0–80% video levels. Even though most network TV stations are synchronized to each other and video timing is very accurate, consumer VCRs may vary in their playback accuracies. They may have quite a bitter of jitter even relative to the color subcarrier.

Typically, you use an analog video sync extraction chip like the LM1881 to find the vertical and horizontal sync positions. You can also extract vertical and horizontal timing directly from video using digital techniques once the signal is digitized with a video ADC.

Before the video signal can be converted to digital, it needs to be conditioned. First, restore the proper DC offset by ensuring the sync pulses are clamped to a known offset. You may want to use an AGC circuit to guarantee consistent video levels.

After the DC offset and level adjustment, low-pass filter it to reduce aliasing in the A/D process. A high-speed flash ADC samples the video at a high data rate (usually four times the color-burst frequency).

Since high-speed ADCs are expensive, this sampling is done with 8- or even 6-bit accuracy. The clock sampling the video signal is usually derived from an analog PLL locked to the color-burst component in the video signal.

Once the signal is digitized, use a counter which locks onto the horizontal sync signal. A horizontal counter is reset whenever the sampled signal goes below the sync threshold value.

To make the counter less sensitive to noise impulses, only reset the counter within a certain window around the time the sync signal is expected. Whenever the counter resets, a horizontal line counter increments.

The line counter gets reset when there's a long sync pulse at the beginning of a horizontal sync period. This synchronizes the vertical counter to the frame rate of the video signal and makes it possible to extract the field number (1 or 2).

There are 525 lines in a frame. Lines 1-262 are field one, and lines 263-525 are field two.

Finding the VITC signal is now a piece of cake. Just reset a shift register at the beginning of the horizontal line where you expect

Bit Position	Description	Bit Position	Description
0	position identifier	40-41	hundreds of days
1–4	seconds units	49	position identifier
5-8	seconds tens	50–58	control function (CF) elements
9	position identifier	59	position identifier
10-13	minutes units	60–68	control function (CF) elements
15-18	minutes tens	69	position identifier
19	position identifier	70-78	control function (CF) elements
20–23	hours units	79	position identifier
25-28	hours tens	80–88	straight-binary-second (SBS)
29	position identifier	89	position identifier
30–33	days units	90–97	straight-binary-second (SBS)
35-38	days tens	99	position identifier
39	position identifier		

Table 3-Since *IRIG timecodes are more generic than the video timecodes (e.g., LTC and WC), they contain data fields in which application-specific data can be encoded (i.e., CF fields).*

the code, and start shifting at the beginning of the VITC signal.

Of course, if you just wanted to build a reader, there's a much simpler way. Since the VITC information is almost symmetric around the ground of an AC-coupled video signal, a comparator can extract the zero crossing of this signal.

Figure 4 shows the schematic of a simple FPGA-based VITC reader. Figure 5 gives you an idea of how the reader works.

The bit clock is synchronized to the edges of this zero-crossing signal. Each bit-clock transition shifts the current state of the video into the shift register.

When all the sync bits match up and the CRC is correct, you know that the received VITC word is correctly framed and the data you're interested in can be converted into ASCII.

Without the CRC, this method is too unreliable since random video may cause the frame-detection circuit to be fooled into thinking that a VITC word has been received.

Bit Position	Description	Bit Position	Description
0	hole	30-33	units of day
1	zero	35-38	tens of day
2	DST indicator	39	position
	leap-second warning	40-41	hundreds of day
437	units of year	49	position
9	position	50	UTC correction
10–13	units of minute	51–55	tens of year
15–18	tens of minute	56	DST indicator
19	position	57–58	UT1 correction
20-23	units of hour		(0.1, 0.2, 0.4)
25-28	tens of hour	59	position
29	position		

Table 4—*IRIG-H* timecodes are similar to *IRIG-{A,B,G}* timecodes, but they use a *100-Hz* carrier and limit the number of bits to 60. This way, one complete frame can be transmitted in 60 s.

The host interface consists of an RS-232 interface. When a valid VITC line has been received, the FPGA transmits hh:mm:ss.ff [where *hh* is hours, mm is minutes, ss is seconds, and *ff* is frame) to the host via its RS-232 interface.

You can build an LTC reader just like a VITC reader. In fact, a combined VITC/LTC reader can be designed by reusing many components. Only a data/clock separator is needed. Framing and direction are determined by the parity bits and sync characters, not a CRC checker.

To build a VITC encoder, lock the VITC generator to the video source. You can do this using one of the techniques I discussed to extract the horizontal and vertical syncs. Just overlay the VITC signal with the video signal.

WHAT'S NEXT?

To edit videos, I have to encode a timecode signal on them. Since they have video and audio, I need a way to add the timecode.

The best way would be to convert the mono audio channel into linear stereo audio, while preserving the original audio on one channel and adding LTC to the other channel. But, there's a problem.

Most VCRs mute their audio circuitry while shuffling the tape so you don't hear an annoying high-pitched squeal when fast-forwarding or rewinding. To use LTC while shuffling on a consumer-grade VCR, the muting circuitry has to be bypassed.

I could overlay VITCs on a dub of the original, but this adds an extra generation to the final copy since it's copied from the dubs.

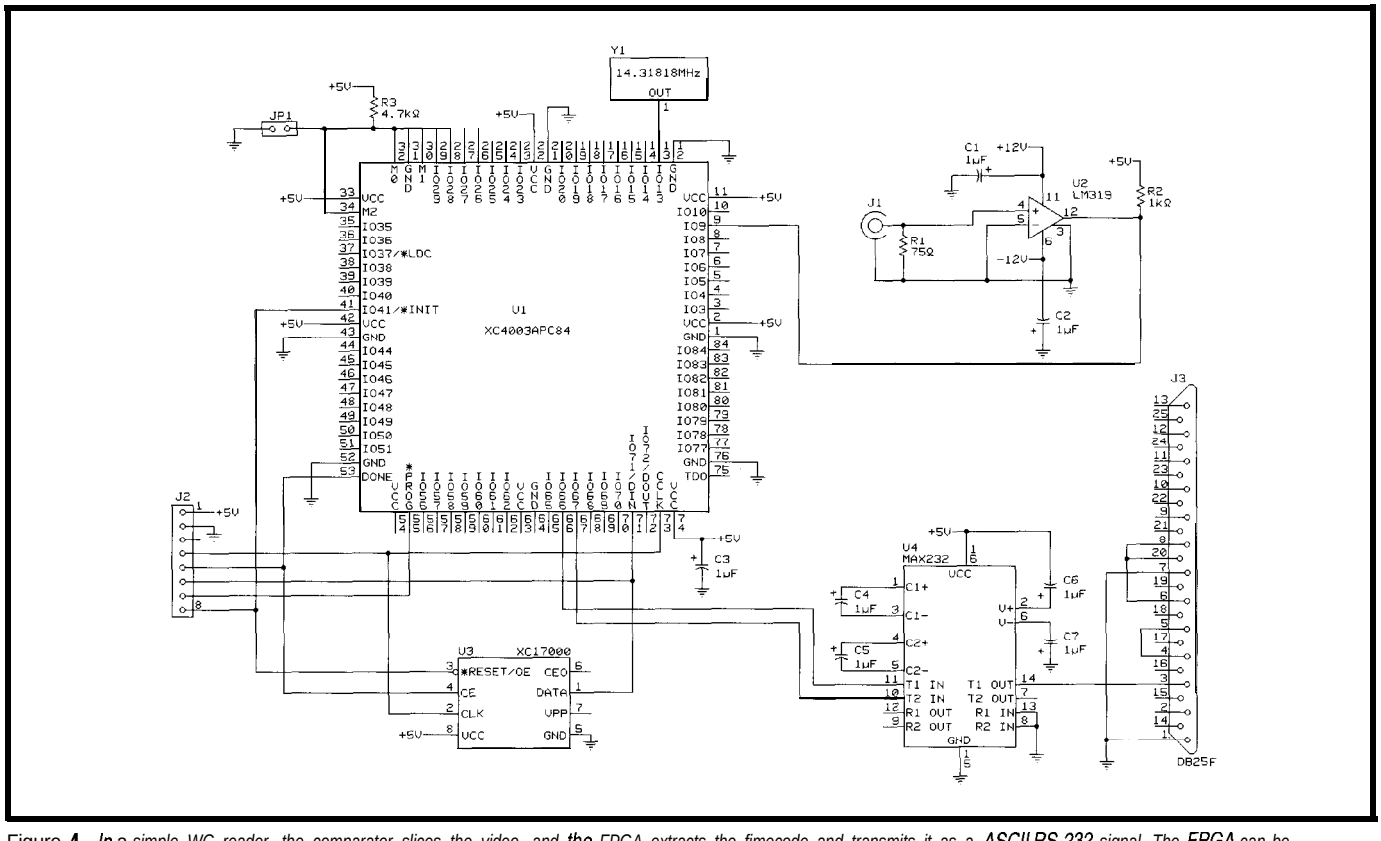


Figure 4—*In a simple WC reader, the comparator slices the video, and the FPGA extracts the fimecode and transmits it as a ASCII RS-232 signal. The FPGA can be configured via serial-configuration PROM or a download tether on (J2) by selecting a jumper (JP1).*

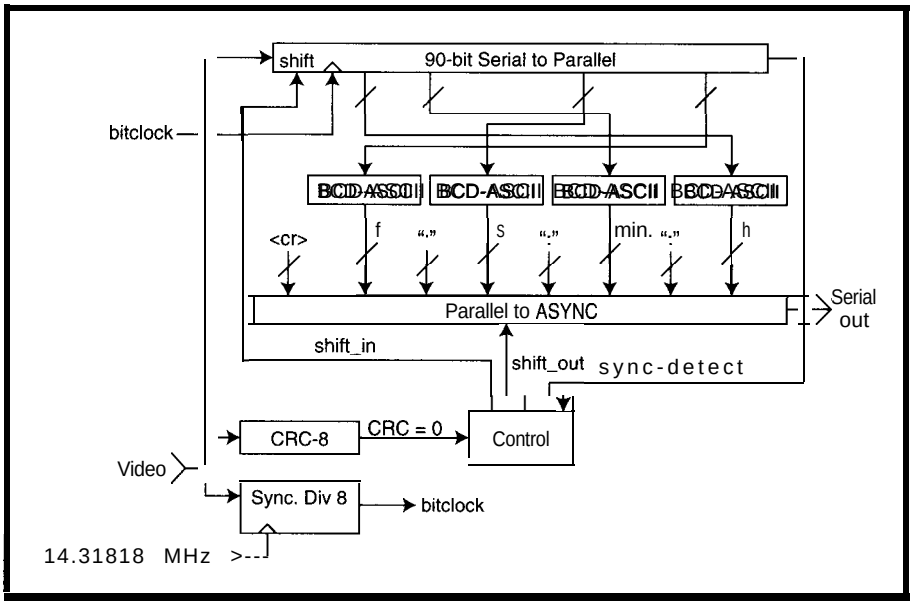


Figure 5—*This* block diagram *shows how the VITC* serial *bitstream is sampled with a* divide-by-eight clock and then *converted to ASCII. The converted* data is sent out via *the serial port (serial_out). The control allows data to* be *sifted in until* a sync *pattern* is defected *(sync_detect) and the* CRC is computed *correctly (CRC= 0). It then* locks *the dafa until if is transmitted via the RS-232 port.*

Once the timecodes have been recorded on the tapes, I can generate an EDL on my computer by simply noting the starting and ending times of scenes I want.

By using the timecode reader to generate the EDL and an IR transmitter interface, I can cue up the source VCR and remotely control the destination deck to record the desired scenes.

The real gem of this technique is that I can use the same EDL to make identical tapes for all the relatives! ❑

Ingo ***Cyliax works in computer sciences at Indiana University where he does system administration and spends a lot of time in the Analog VLSI and Robotics Lab. You may reach*** *Ingo* ***at*** *cyliax@* ***EZComm.com.***

REFERENCES

K. Jack, ***Video Demystified: A Handbook for the Digital Engineer,*** HighText Publications, Solana Reach, CA, 1994.

J. Watkinson, ***The Art of Digital Video,*** Focal Press, Oxford, U.K., 1994.

Xilinx, Inc., ***The Programmable Logic Data Book, 1994.***

I R S

401 Very Useful
402 Moderately Useful
403 Not Useful

FEATURE ARTICLE

Peter Sorrells & Shannon Poulin

Converting VGA Monitors

Plain-Vanilla Monitors Become Plug-and-Play VESA-Compatible Designs

With the rapidly increasing popularity of plug-and-play capability, manufacturers want to convert their monitors into DDC-compliant ones. Peter and Shannon offer a single-chip conversion solution.

Since the advent of the IBM PC and PC compatibles, users have contended with manual configuration of peripherals and plug-in cards. DIP switches, special software drivers, technical reference manuals, and manual software configuration make it difficult to configure a new PC or add a peripheral.

With plug-and-play peripherals, you simply plug in a part and reboot the system. Configuration is accomplished automatically and transparently.

With Windows 95, Microsoft is moving this direction. Standards committees (e.g., JEDEC and VESA) are building uniform interfaces. PC and peripheral manufacturers are designing plug-and-play products. New DRAM DIMM modules have this capability.

In this article, we look at the design considerations for converting a standard VGA or SVGA monitor into a DDC-compliant plug-and-play monitor.

WHY CONVERT?

Approximately two and a half years ago, the VESA committee began creating the Data Display Channel (DDC) standard, which includes specifications for data format and the communication bus.

In 1995, many market leaders offered DDC-compliant monitors in their product lines. Graphics chip-set companies like S3 and Cirrus Logic had VGA/SVGA chip sets for the DDC bus, and PC manufacturers provided DDC-compliant systems. The volume of DDC-compliant monitors and systems exploded from zero to millions of units per month, causing a fundamental shift in user expectations.

With plug-and-play capability (DDC in monitors), installation time is greatly reduced. What took over an hour can frequently be done in minutes.

As more people use plug-and-play peripherals, demand will swell. Older-style peripherals just won't be bought. To stay in the monitor business, it'll be necessary to provide plug-and-play. Eventually, DRAM DIMMs, printers, mice, fax/modems (won't that be nice!), and other peripherals will need it.

There's approximately $10-20 difference between a "vanilla" SVGA monitor and a DDC-compliant one, all specs being equal. If the conversion is accomplished properly and efficiently, the cost of additional components is much less than the retail differential.

A monitor manufacturer gains a greater margin due to the added feature set and stays in business by offering the unit in highest demand. Just as CGA and EGA monitors fell to the superior technology of VGA and then SVGA, monitors without plug-and-play capability may soon become dinosaurs.

VESA EDID TABLE

For a video-graphics adapter in a PC host to properly communicate with a monitor, several characteristics of the monitor must be taken into account—

Section	# Bytes
Header string	8
Vendor/Product information	10
EDID version, revision	2
Basic display parameters	5
Color characteristics	10
Established timings	3
Standard timing identification	16
Detailed timing description	72
Extension flag for future extended tables	1
Checksum	1

Table I--The *EDID table stores critical monitor parameters for transmission to the host.*

resolution, horizontal and vertical image size, feature support, color characteristics, and timing.

The VESA DDC specification includes a detailed Extended Display Identification Data (EDID) table which stores monitor characteristics in a standard format. In this manner, all monitor, PC, and video-graphics manufacturers use a common frame of reference for storing and communicating critical information.

The EDID table isn't large—only 128 bytes-but it stores the necessary information for most monitors. It's organized into the basic sections shown in Table 1. A complete specification of the EDID table can be found in the VESA EDID standard.

Because of the detailed information provided in the EDID table, any monitor carrying this data can be instantly connected to a host communicating across the DDC bus. Information from the table automatically loads into the host graphics adapter and properly configures the monitor. The entire operation is transparent to the user.

Another data table, the Video Display Identification Format (VDIF), provides extra information not included in the EDID table. However, most DDC-compatible monitors use only the EDID table.

Complete specifications of the VDIF table are found in the VESA VDIF standard. In this article, we only address the conversion of standard VGA/SVGA monitors to DDC-compatible models using the basic EDID table.

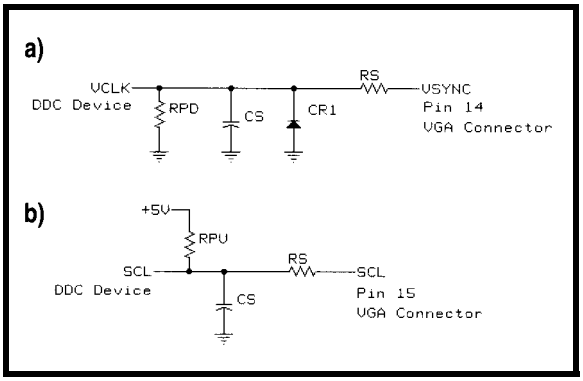


Figure 1 a-A *simple external filter conditions the !/sync signal for the DDC device. b-The SCL line uses a simple RC filter to attenuate transients and prevent false* switching.

DDC COMMUNICATION MODES

The EDID table information is stored in a nonvolatile device inside the monitor and is transferred to the host at powerup. Two unused pins in the standard VGA connector are redefined to provide signal paths for one clock and one data line for communication of the EDID table-pin 12 for data and pin 15 for clock (DDC2 hosts).

Four basic modes of communication are identified in the VESA standard—DDC1, DDC2B, DDC2B+, and DDC2AB.

DDC1 mode is the most basic communication mode. It consists of a single clock line and single data line. The Vclk line is generally connected directly to Vsync and therefore runs at a relatively slow rate. However, it can be ramped by the host to a higher frequency (up to 25 kHz) while transferring data, then slowed to its normal 60–120-Hz rate.

In DDC1, the EDID table is continuously fed to the host and repeated indefinitely. The host only has to look for the header bytes to synchronize itself to the EDID datastream. At powerup, all DDC-compliant monitors must be in DDC 1 mode.

In DDC2B mode, the host takes control of the communication bus and sends its own clock (SCL, VGA connector pin 15) to the monitor, separate from the Vsync clock. The presence of this clock causes the monitor to switch communication modes from DDC 1 to DDC2B. The monitor then operates in I²C protocol.

As a DDC communication bus, it enables clock rates up to 100 kHz without disrupting Vclk and enables the host to acquire specific data. Because every byte of data in the EDID table resides in a known location, the host can request and receive only the bytes it needs rather than waiting for the entire table to cycle through.

DDC2AB signifies an ACCESS.bus host and ACCESS.bus monitor, using a bidirectional bus which can operate on a full ACCESS.bus system. The hardware layer for ACCESS.bus is the I²C bus itself, but certain addresses and commands in the ACCESS.bus specification may cause problems in a DDC2B-only monitor.

A host can be a DDC1, DDC2B, DDC2B+, or DDC2AB type. A DDC2B+ host can translate from ACCESS.bus to I²C by bit banging the monitor in a bidirectional bus mode. However, it doesn't require the monitor to support the rest of the ACCESS.bus command set. In other words, in a DDC2B+ system, the monitor is simply DDC 1 and DDC2B compatible.

MONITORS

All DDC-compliant monitors must support DDC 1 and DDC2B modes but not necessarily DDC2AB. Because all peripherals attached to an ACCESS.bus system must be capable of becoming a bus master and interpreting the ACCESS.bus command set, a DDC2AB monitor is normally required for an ACCESS.bus system.

Some DDC2B monitors, however, function properly in an ACCESS.bus system if the host has DDC2B+ capability and only sends I²C commands to the monitor. All DDC-compliant monitors must switch from DDC 1 mode (power-up, basic communication mode) to DDC2B mode (1°C) immediately on detecting a DDC2B host. This task is accomplished by monitoring the SCL clock line from the host.

A DDC1/2B (DDC-compliant) monitor supports DDC1, DDC2B, and DDC2B+ hosts,

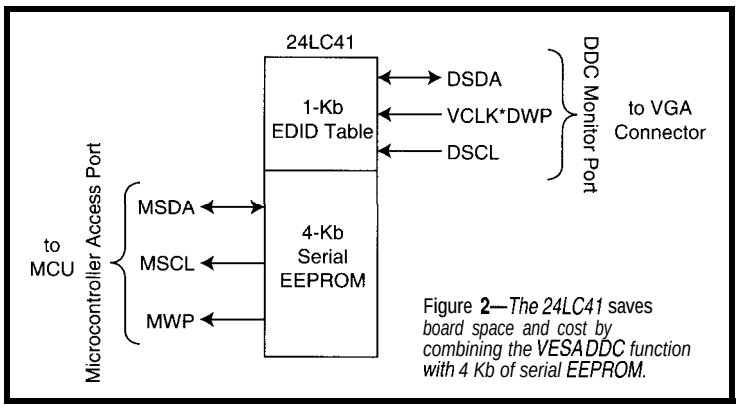


Figure 2—*The 24LC41 saves board space and cost by combining the VESA DDC function with 4 Kb of serial EEPROM.*

24LC21	1-KB DDC Serial EEPROM
24LCS21A	I-KB DDC Serial EEPROM with Error Correction and Write Protect
24LC41	1 -KB DDC Serial EEPROM plus 4-KB MCU EEPROM
24LC41A	1 -KB DDC Serial EEPROM with Error Correction plus 4-KB MCU EEPROM

Table 2—*Cost and performance requirements can be matched to a* number of available *devices.*

whereas a DDC2AB monitor supports a DDC2AB (ACCESS.bus) host.

HOSTS

Four types of hosts can be presented to the monitor.

A DDC1-only host only uses DDC1 mode and does not present an SCL clock to the monitor. The monitor must use Vsync to continuously clock data to the host.

A DDC2B host provides an SCL clock to the monitor and requests specific data by address and at a much higher rate of speed. The monitor must immediately detect this and provide data at the very first command word.

A DDC2AB host is an ACCESS.bus host which may provide ACCESS.bus commands to the monitor. It requires a DDC2AB monitor.

A DDC2B+ host is an ACCESS.bus host that only presents pure I^2C bus commands to the monitor. It doesn't require a DDC2AB monitor-only a DDC-compliant monitor.

Some design considerations are prudent due to the high-noise environment inside the monitor and the reality that users will hot-plug the monitors and cycle power to the host and monitor in unpredictable ways.

SIGNAL CONDITIONING

The Vsync signal is relatively clean, synchronizes the monitor's vertical scan, and sequentially clocks EDID data from monitor to host in DDC1 mode. But quite often, it carries high-voltage transients and other noise that confuse the monitor's DDC circuitry, which stores and transmits EDID data.

Transients and noise are sometimes induced by the host graphics card or other monitor components. Cleaning up this signal goes a long way toward eliminating switching problems between communication modes and avoiding bit errors in transmission.

Figure la shows a circuit successfully used by many monitor manufacturers. CR1 eliminates under-voltage transients and the RC filter helps shunt unwanted high-frequency noise and high-voltage transients.

A pull-down resistor is required because Vsync is capacitively coupled to other circuits inside the monitor. During a short-term powerdown, these capacitors partially discharge, giving an analog voltage which may be between the DDC device's V_{IH} and $V_{,,}$. This can confuse the state machine of some DDC hardware, preventing the proper DDC1–DDC2B start-up sequence at powerup.

The original VESA DDC standard required the monitor to remain in DDC2B mode until reset by a powerdown condition. The DDC circuit may

not completely reset if voltage is continuously applied between powerup and powerdown. The pull-down resistor causes a rapid discharge of the series coupling capacitors so that the DDC hardware begins from a reset condition at powerup.

COMMUNICATION ERRORS

The original VESA DDC standard requires the monitor and host to be powered up together. In practice, however, users randomly power the host and/or monitor up or down at any time and in any sequence. Because the standard requires the monitor to switch from DDC1 to DDC2B mode on any falling edge detected on the SCL clock line, communication errors can occur under several conditions.

One condition occurs when the host and monitor are already communicating in DDC2B mode and the host is powered down. When it is powered up again, it may expect a DDC1 bitstream from the monitor. The monitor, however, is still in DDC2B mode, waiting for a command byte.

There is virtually no way to avoid this issue electronically without violating the original DDC standard, which states the monitor must remain in DDC2B mode until powerdown.

Providing a manual reset on the monitor is an option, but it isn't popular because of the added cost and change of philosophy. Plug and play means no human intervention, even if, as in this case, human intervention caused the error!

Communication errors also occur when the host and monitor are hot-plugged together. The voltage transients caused by hot-plugging may result in a negative-going edge on SCL, which switches the monitor immediately into DDC2B mode.

If the host is a DDC1 host, it will not receive any configuration information from the monitor. The filtering circuits in Figure lb help reduce this problem.

Errors also occur if the monitor is disconnected from one host and connected to another while power is applied. This combines the other two conditions.

If the new host is a DDC1 host and the monitor is in DDC2B mode, the host won't receive any EDID information due to a voltage transient on SCL or prior communication in DDC2B mode. The filtering circuits in Figure lb reduce the voltage transient effect.

ERROR RECOVERY

The original VESA DDC Standard V. 1 .O requires the monitor to switch from DD1 mode to DDC2B mode on any falling edge of the SCL clock. Ideally, when the host and monitor are not hot-plugged or disconnected while operating or powered up at different times, the criterion for switching communication modes is adequate. In the situations just described, however, fatal communication errors are possible.

When VESA revised the standard to V.2.0, it upgraded the switching requirements and provided for more intelligence in the monitor's DDC controller. Now a DDC monitor may test the host!

A DDC 1 -only host never sends clock pulses onto the SCL line. A DDC2B, DDC2B+, or DDC2AB host sends clocks onto the SCL line but also provides valid I²C control bytes onto the data bus (SDA, pin 12 on the VGA connector). The DDC circuitry in a new DDC-compliant monitor on receiving a falling-edge on SCL may have the intelligence to check for a valid control byte before going into DDC2B.

Two options are presented for error recovery-checking for a control byte during 128 Vsync clocks or for 2 s of SCL quiet. Either method results in a return-to-DDCl mode if no valid control byte is received.

This new standard creates a much more robust system that is essentially immune to noise transients, voltage spikes, and other spurious events.

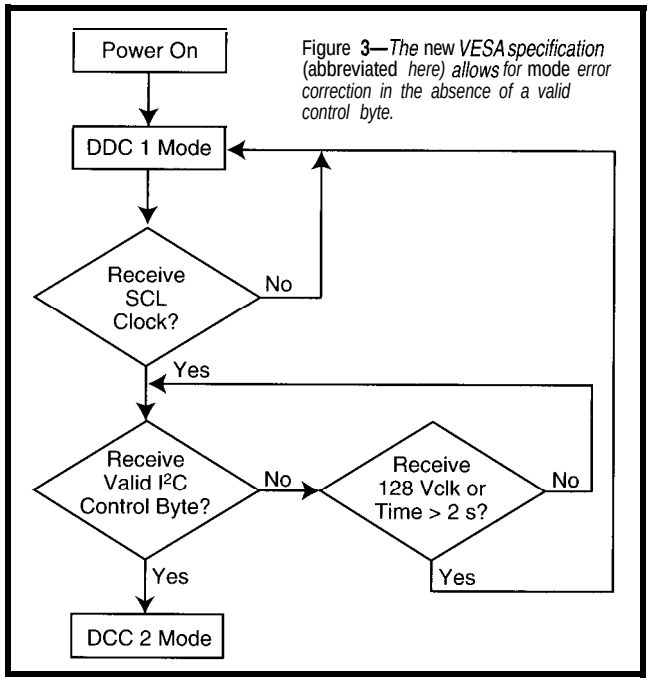


Figure 3—*The new VESA specification (abbreviated here) allows for mode error correction in the absence of a valid control byte.*

CONTROLLER-BASED DESIGNS

The first DDC-compliant monitor prototypes orchestrated data transfer between monitor and host via general-purpose microcontrollers. In fact, ACCESS.bus monitors still do.

The beauty of a field-programmable microcontroller in comparison to a custom silicon solution, of course, is its flexibility.

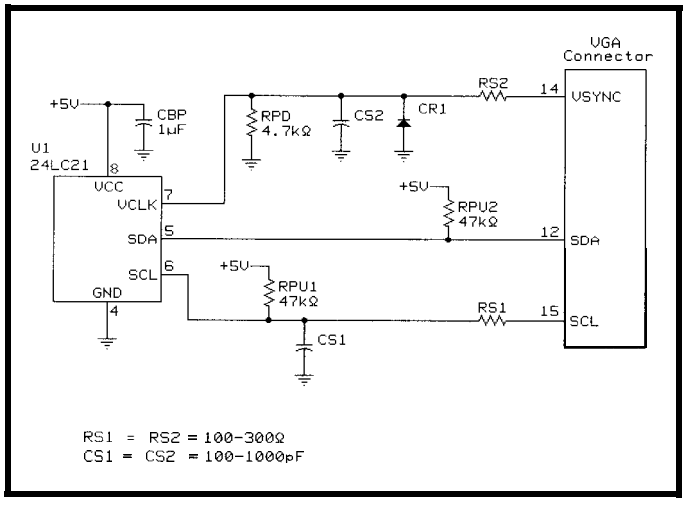


Figure 4—*Wiring the 24LC21 and ifs few external components to the monitor's VGA connector is simple. No other monitor circuitry is affected.*

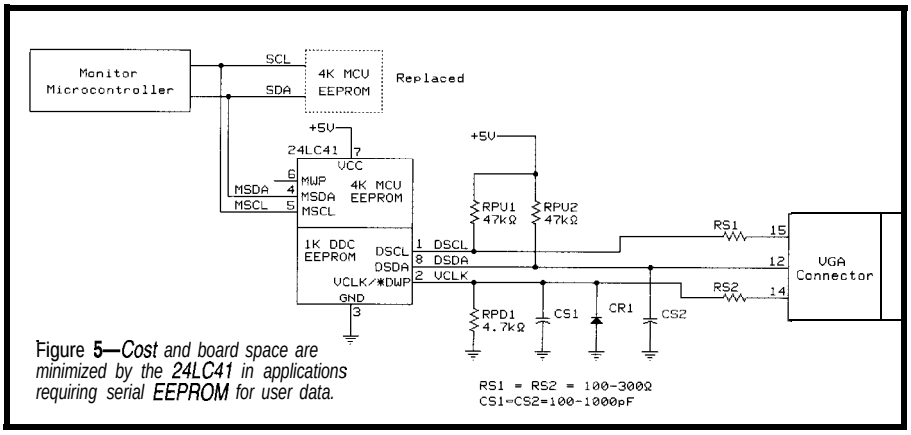


Figure 5—*Cost and board space are minimized by the 24LC41 in applications requiring serial EEPROM for user data.*

Changes in customer requirements, anomalies in particular video-graphics chips in the host, and nearly any surprise can be made up for in software. Its disadvantages are size and cost.

Although monitors are cavernous inside, their circuit boards are compact to keep cost down and address RF interference issues. A microcontroller in an 18-lead package and the external serial EEPROM to store the EDID table itself can be quite difficult to shoehorn into some monitors.

As mentioned, the retail difference between vanilla versus DDC-compliance for a 14–15″ monitor is about $10–20. After including distributor and retail markups, a $3–5 microcontroller solution provides virtually no margin at all. It only helps the manufacturer sell the monitors.

Higher-end monitors with advanced features (e.g., ACCESS.bus) command a much higher price tag and usually have a microcontroller or two inside them. Sometimes, the existing microcontroller can be configured to control the DDC bus in addition to other tasks.

USB may have some effect here. Some higher-end monitors will use it for EDID and VDIF communication.

However, the new extended VGA connector still includes a DDC communication channel. And, the cost of a dedicated DDC-chip solution is much lower than the controller solution.

INTEGRATED DESIGNS

Dedicated microcontroller solutions are available from SGS-Thomson and Motorola. Both companies provide dedicated microcontrollers for mid-range and high-end monitors. Now, their portfolio includes those with onboard DDC data storage and external pins for DDC communication.

However, the highest-volume monitors are still in the 14-15" low- to medium-cost families. Some of these now use an onboard microcontroller and serial EEPROM to control the user interface and on-screen display device. But, dedicated microcontrollers from any supplier are several times more expensive than a dedicated silicon solution whose sole purpose is the storage and communication of the EDID table.

SINGLE-CHIP SOLUTIONS

Microchip Technology introduced the first dedicated DDC device for monitors shortly after the VESA DDC standard was released. The 24LC21, named for its two clocks and 1 Kb of memory, was designed to meet the original standard.

This successful device is still designed into most DDC-compliant monitors. It became the de facto standard and was copied by several other silicon manufacturers.

Although the 24LC21 met the complete VESA V. 1 .O standard, the noise issues associated with the V. 1 .O standard surfaced during the spring 1995 VESA Plugfest. During this test, scores of monitors and PC hosts were intermingled to determine compatibility.

The hot-plugging issues, power transients, and Vsync noise issues became apparent as the results rolled in. At this time, the signal-conditioning circuits in Figure 1 were first designed and implemented. With the altered circuit design, the 24LC21 is still used in many monitors.

The 24LC41 was also designed for the V.1.0 specification. Some monitors already require a serial EEPROM for user functions in the monitor—usually 2 or 4 Kb—and much investment has already been made in code for the existing microcontroller. The 24LC41 can be added to the system without changing any code.

It contains both a DDC device with dual clocks and a separate DDC bus and a 4-Kb standard serial EEPROM with its own I²C bus. The 4-Kb half of the 24LC41 connects exactly where the old serial EEPROM was connected (assuming it was a 2–4-Kb I²C standard device). The DDC half of the 24LC41 connects directly to the VGA connector as shown in Figure 2.

However, the release of the VESA DDC Standard V.2.0 allowed the upgrade of the 24LC21 to create a newer, smarter device—the 24LCS21A. The 24LCS21A incorporates error recovery by counting Vsync clocks as it searches for a valid I²C control byte.

Its state machine follows the VESA flowchart shown in Figure 3. It also incorporates a write-protection pin for data integrity as well as special addressing tweaks (it only wakes up to address 000) to minimize any potential problems if connected to an ACCESS. bus or DDC2B+ host.

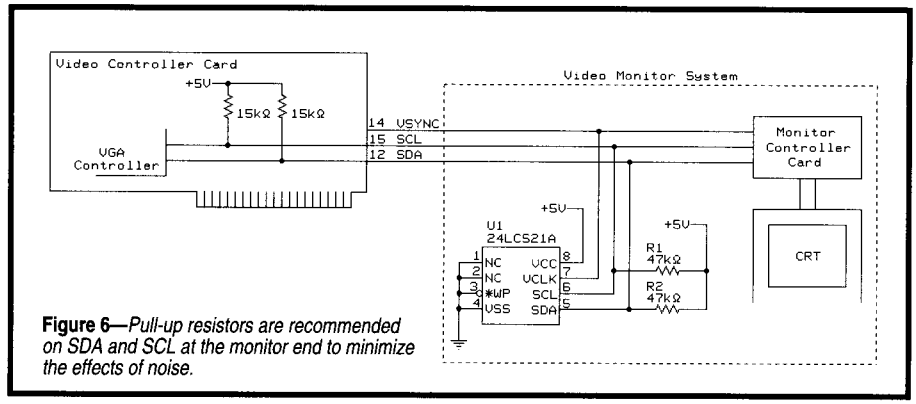


Figure 6—*Pull-up resistors are recommended on SDA and SCL at the monitor end to minimize the effects of noise.*

Best of all, the filter circuits in Figure 1 are pulled inside the 24LCS21A. Schmitt triggers and onboard filters virtually eliminate the need for external components.

An available dual-ported device, the 24LC41A, is downward compatible with the 24LC41. The 24LC41A includes the filtering and error recovery of the 24LCS21A. In most applications, the 24LCS21A is also downward compatible to the 24LC21. Since external components are eliminated, it instantly upgrades performance at no extra cost.

These dedicated DDC devices are very efficient in real estate, price, and design time. They are available in 8-lead SO (JEDEC, 150-mil) and 8-lead PDIP packages, cost less than $1, and can be designed in within an hour.

The entire subsystem is on-chip and virtually all connections go to the VGA connector. No software changes

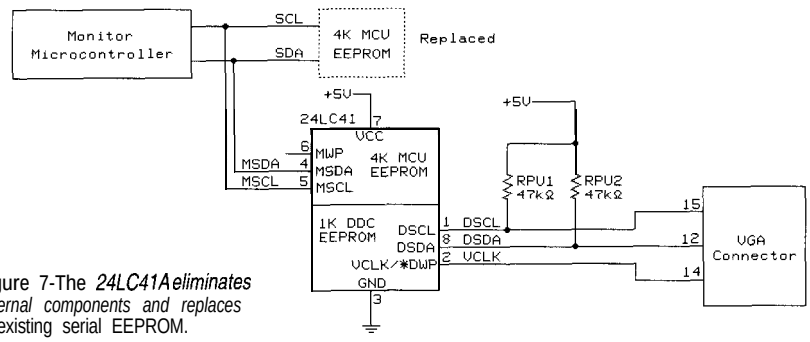


Figure 7-The *24LC41A eliminates external components and replaces an* existing serial EEPROM.

or microcontroller upgrades are necessary.

Connection to practically any existing monitor design is quick and easy, using proven schematics and only a few external components.

The DDC devices are listed in Table 2.

UNIVERSAL APPLICATION CIRCUITS

Figure 1 a and b combine in Figure 4 to form a complete application circuit for a 24LC21-type device. External filtering is necessary because the 24-LC21 meets only VESA DDC Standard V. 1 .0. Still, this option is very cost effective and time proven.

Figure 5 shows the application circuit for a 24LC41 dual-ported device in a monitor with an existing microcontroller. Figures 6 and 7 provide application circuits for the 24LCS21A and 24LC41A, respectively. The circuits are much simpler because all filtering is brought inside the chip.

Gaining an extra $10-20 at retail, revitalizing an existing product design to match new technology, and moving into a new paradigm has never been this easy or this inexpensive. ❑

Peter Sorrells ***is a strategic marketing*** manager for Microchip's Memory ***and Specialty Products. He received a BSEE*** from the University of Arizona ***and holds three patents. You may reach Peter at*** *pete.sorrells@microchip.com.*

Shannon Poulin is an applications engineer for Microchip's Memory and Specialty Products. He holds a BSEE and MBA from the Florida Institute of Technology. You may reach Shannon at *shannon.poulin@microchip.com.*

SOURCE

24LC*xx*
Microchip Technology
2355 W. Chandler Blvd.
Chandler, AZ 85224
(602) 786-7200
Fax: (602) 899-9210

I R S

404 Very Useful
405 Moderately Useful
406 Not Useful

Image Filtering Theory

What's a Convolution Filter?

Removing noise from digital images is often regarded as a black art. Bruce discusses his mighty magic using filtration, aliasing, Fast Fourier Transforms, and convolution codes. It's all a matter of the stir.

FEATURE ARTICLE

Bruce Hubbard

Most of us encountered the Fourier transform (FT) in an engineering class of some sort. Whether you understood the math or not, you probably came away with the idea that an arbitrary wave can be assembled by adding combinations of simple sines and cosines.

In Figure 1, I added the first five Fourier components of a square wave and obtained a wiggly, but squarish-looking profile. As more components are added, the approximation to a square becomes better and better.

What most people don't realize is that the same ideas apply to 2D images and lie at the heart of image-processing theory.

ORTHONORMAL BASIS

To understand image transforms, let's start with a vector-based analogy. Any vector *V* can be expressed as the sum of one or more vectors, which are then said to form a basis for *V*.

For example, an arbitrary 3D vector *V* can be projected onto three vectors, x, y, and z, using the dot product—*V* • *x*, *V* • *y*, and *V* • *z*. *V* can be reconstructed by adding these projections back together using ordinary vector algebra.

Any three vectors can be used, but selecting a normalized orthogonal set means that the results are simpler and that *V* is represented by a unique sum of the vectors. A basis is complete if any vector can be represented as a scaled sum of the basis vectors.

A similar process is shown in Figure 1, where a square wave is projected onto an infinite set of sine and cosine functions. The result turns out to be a scaled set of sine functions. The lowest frequency component is one-half of a sine wave, followed by components of increasing frequency in the sequence 3x, 5*x*, and so on.

BASIS IMAGES

Two-dimensional image transforms are analogous to vector projections and FTs. It's possible to construct a general orthonormal basis for any 2D image and represent the image as a scaled sum of these basis images.

For example, the 64 images shown in Figure 2 form what is known as a Hadamard basis. It is complete as these 64 are sufficient to construct any 8 x 8 grey-scale picture. The red lines simply help to separate adjacent images.

The lowest-frequency component is in the upper left, with frequency (here called sequency) increasing from left to right and top to bottom. Each basis component is orthogonal-it cannot be made by combining any of the others.

How these components combine to form an image is shown in Figure 3, where five of the major components of a diagonal line are appropriately scaled and added together. A white base image with a magnitude of 1399 is folded into the first summation. As the sequency of each component increases, finer and finer detail is realized.

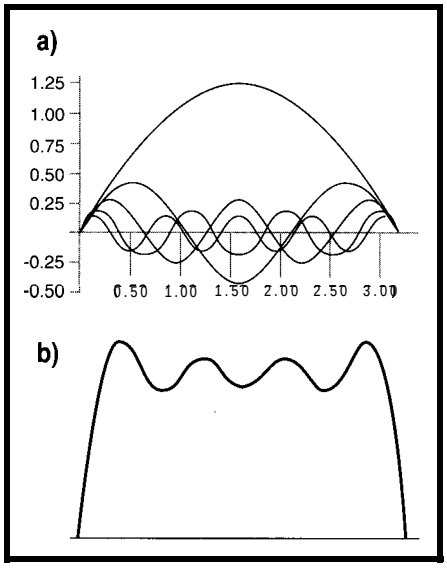


Figure I--Nearly any shape can be *constructed by* combining *simple* sinusoidal *curves.a—The* first five components *of the* Fourier *expansion of a square wave can be added together to form the rectangular profile in (b). b-Adding more components makes the rectangle increasing/y perfect.*

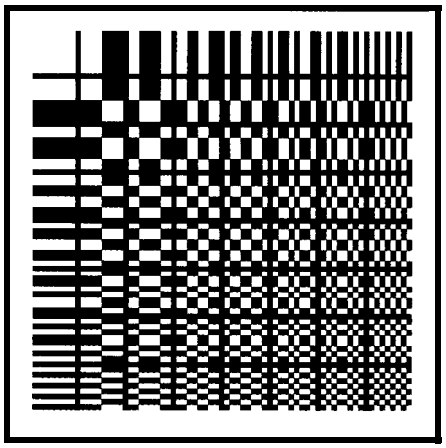


Figure 2—*Any* image can be *constructed* by adding together components of the Hadamard basis, each scaled *by the* appropriate signed real *number. These 64 components suffice for* constructing *any 8 x 8 image. White* represents + 1, and black represents -1. Red lines are *simply* visual separators.

I selected the Hadamard transform for this example because its basis functions are easy to display. Its main claim to fame is ease of computation. Its main drawback is that it doesn't resolve an image into a small number of components.

All 64 components are needed for this simple example-the remaining 59 cancel out each other's undesired high-frequency contributions. Their absence results in the speckled background.

Small projection spaces make it easier to use transforms to compress images by omitting minor components and to filter noise out of images by removing their projection.

FTs also suffer from large projections, as well as the added complications of complex arithmetic. Note that FTs are periodic and require an infinite number of components. The Hadamard transform is finite and not periodic.

So, in choosing basis functions for image processing, look for a set that expresses all the energy of a picture in a small number of components. One popular choice is the cosine transform. Another option is to use wavelets, which are spatially localized basis functions.

FILTERS IN ONE DIMENSION

This brings us to filtration, which is the heart of the matter. Images are usually filtered for one of two reasons—to remove noise or to change contrast (i.e., softening or sharpening).

Filter operation is easy to understand when described in terms of an image spectrum. Filters selectively modify frequency components. Softening filters attenuate high frequencies, sharpening filters attenuate low frequencies, and noise removal deletes selected undesired frequencies.

Let's start with a look at a 1D FT example. A pure sawtooth profile containing all frequency components is shown in Figures 4a (frequency profiles) and 4b–d (density profiles).

Figure 4b is an unmodified sawtooth. In Figure 4c, the high frequencies have been attenuated, thereby broadening and lowering the saw peaks. The resulting image looks softer, and the light-to-dark transition is broader. The lower graph shows the amplitudes of the frequency components before filtration [black) and after filtration (red). Notice how the higher frequency coefficients are attenuated.

In Figure 4d, the lower frequencies are attenuated, resulting in a nearly flat density profile. The grey-scale ramp has all but disappeared, while the light-to-dark transition is enhanced.

The banding-or ringing-is an artifact of excessive frequency removal. It should be minimized because it detracts from the image and looks amateurish.

You can see that the light and dark juxtapose at each boundary. This is called the Gibbs effect and results from missing frequencies. It provides enhanced visual contrast.

Thus, slowly changing information (e.g., the grey ramp) corresponds to low frequencies and is enhanced by low-pass filtration. Edges correspond to high frequencies and are enhanced by high-pass filtration.

CONVOLUTION

Two main techniques are used to filter image data. You can compute a frequency transform, multiply the resulting spectrum by a frequency filter, and transform back, which is the approach I used for Figure 4.

The other technique is to perform a convolution of the image with the filter. The convolution theorem states that the convolution of two images is equal to the product of their FTs. In

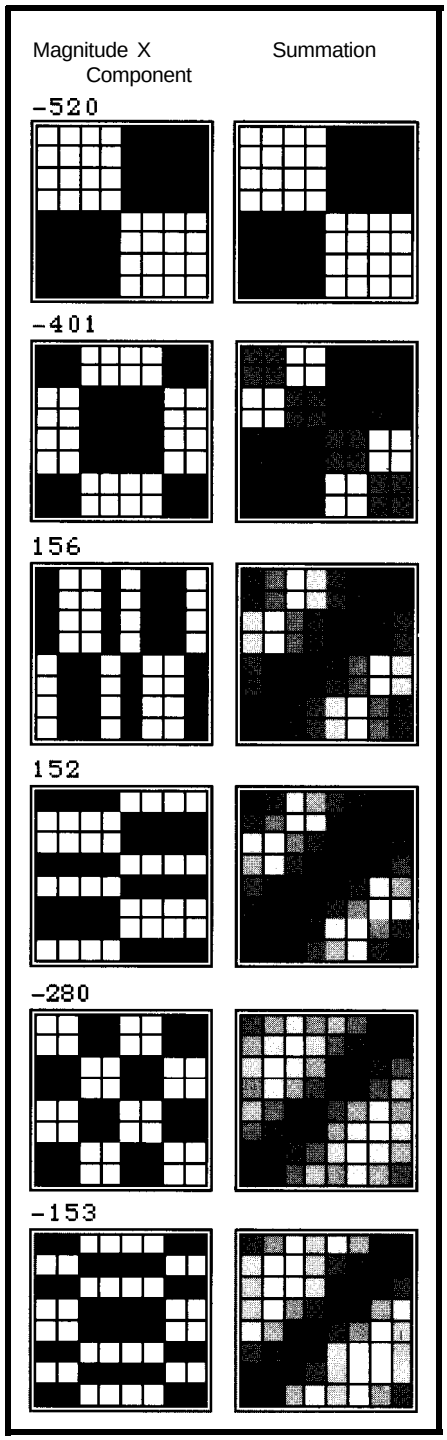


Figure 3-Six *of the* major Hadamard components of a diagonal *line are combined. The component and its scaling coefficient are shown on the left,* and *the* sum is *on the* right. *Each row shows the effect of summing that additional component. All 64 components are required for perfect reconstruction.*

one dimension, the Fast Fourier Transform (FFT) is competitive with convolution. In two dimensions (i.e., images), convolution is generally faster.

The filter, which is a small (3 x 3 to 5 x 5) array, is called the kernel. This kernel is translated over the image as shown in Figure 5. At each point of the

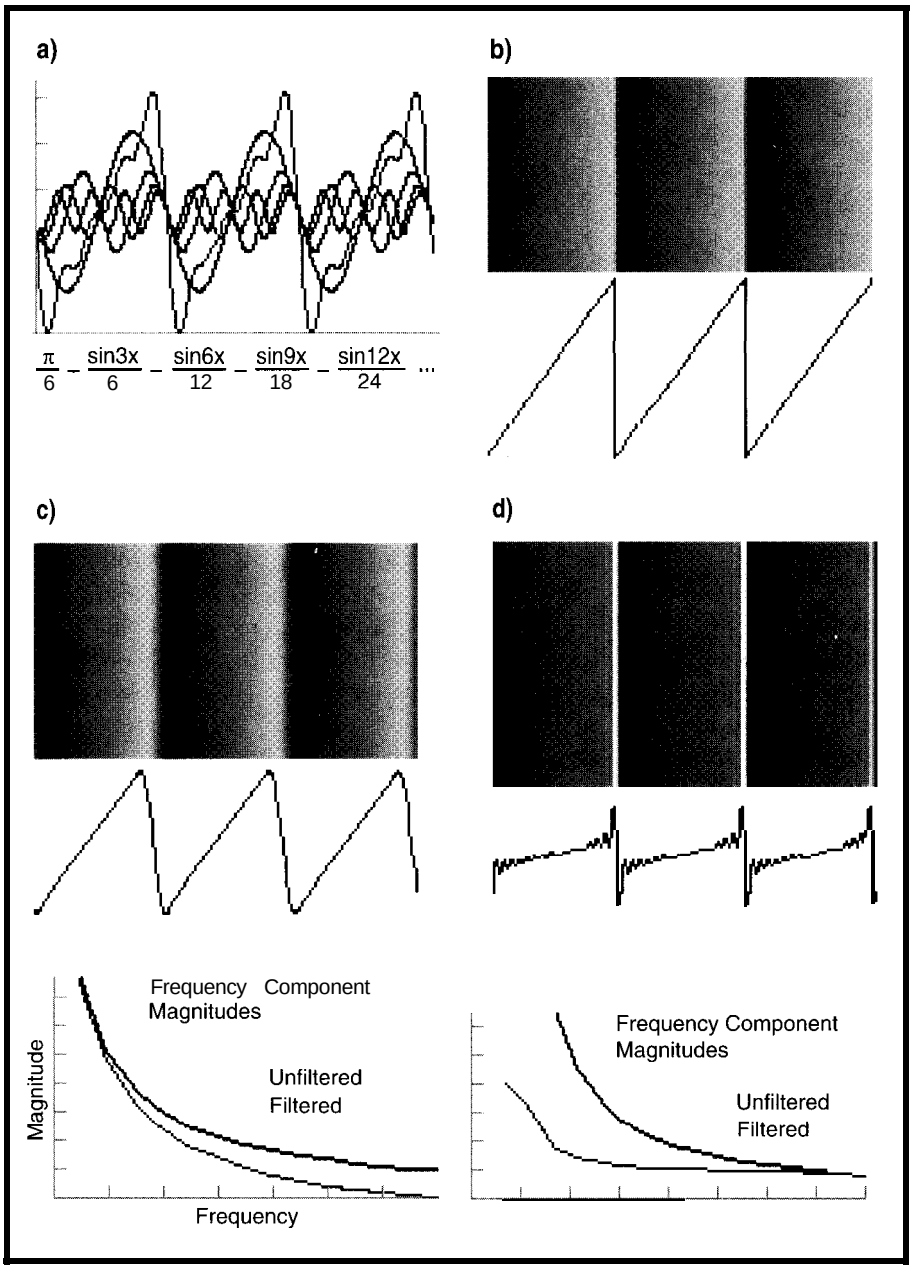


Figure 4a—*The* first four *Fourier components of a sawtooth profile are shown in black. Their summation is overlaid in red.* ***b****—This 2D image corresponds with the triangular profile in (a).* ***c****—The effect of low-pass filtration of(b) results from the selective attenuation of the high-frequency components in (a).* ***d****—The effect of high-pass filtration of* ***(b)*** *results from the selective attenuation of ifs low-frequency components. The density profiles are in the middle graph, and* fhe attenuation of *the* frequency *components is* in *the* bottom curves.

image, the corresponding kernel and image points are multiplied together, summed, and scaled.

The results are usually scaled by the sum of the kernel weights. This sum becomes the point of the output image that lies over the kernel's center.

You can reduce the number of multiplications by taking advantage of kernel symmetries and caching its row and column sums. Convolution may be efficiently implemented in hardware by cylindrically shifting image lines past the kernel.

Finally, convolution is associative $(a[bc] = [ab]c)$, commutative $(ab = ba)$, and distributive over addition $(a[b + c] = ab + ac)$.

FILTER KERNELS

The most basic kernels are the so-called ideal filters. These rectangular shapes in the frequency domain are shown in Figure 6a. Programmers often prefer these filters because of their simplicity.

But, the FT of a rectangle is the sinc function (i.e., $\sin(x)/x$) which oscillates

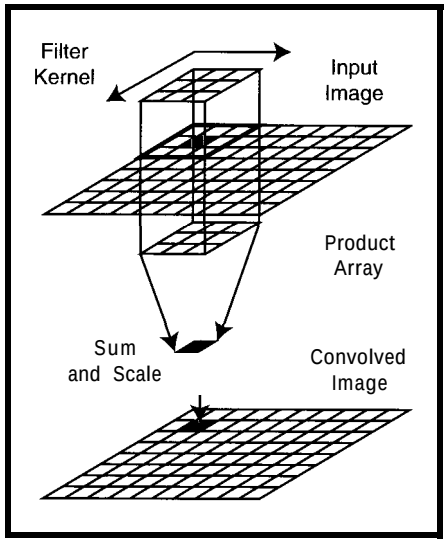


Figure 5—*In the convolution process for 2D images, the filter kernel is normally an n x n rectangular array of numbers. The kernel is laid over the image, and each kernel weight is multiplied by the pixel value directly below it. The resulting n² products are summed, scaled by the sum of the kernel weights, and used to generate one output pixel (in a separate image), corresponding in location to the center of the filter kernel. The filter kernel is translated by one pixel, and the process is repeated over the entire input image. This process is equivalent to multiplying the Fourier transforms of the image and kernel.*

between positive and negative up to infinity (see Figure 6a). This oscillation subtracts adjacent frequencies (as an FT) or pixels (as a convolution). The resulting sums can go negative, causing artifacts such as ringing and polarity reversals in the image.

If you must use a rectangle as a low-pass filter, make it no more than two pixels wide. It's better to use a triangle. Its FT is $\text{sinc}^2 x$ and is always positive.

The width and height of a convolution kernel is inversely proportional to the width of its FT (see Figure 6a). Wide-frequency boxes correspond to narrow pixel sincs and vice versa.

Gaussian profiles work much better for constructing filters because the FT of a gaussian is also a gaussian (see Figure 6b), and these die out smoothly. Several gaussian filters profiles are shown in Figure 7.

The high pass with roll-off is useful for sharpening while minimizing ringing. It may be assembled by subtracting two gaussians (see Figure 8a). The FT of this filter is shown in Figure 8b. The inverse similarity between the transformed gaussians changes the profile of G_1(S)-G,(S) from two peaks in Figure 8a to the famous "Mexican hat" profile of Figure 8b.

Several rules can be applied to gaussian filters. For low-frequency filters, G(0) represents the gain multiplier for large low-contrast areas of the image.

If G(0) = 1, there's no change. If it's greater than 1, contrast decreases (blurs), and if it's less than 1, contrast increases. For high-pass filters, G(0) = $Area(g_1)$– $Area(g_2)$ and $G_{max} \leq Area(g_1)$.

These ideas generalize naturally to two dimensions. Here, convolution kernels are represented by small matrices of numbers as shown in Figure 9. Each kernel is scaled and offset so that it is normalized to 1.

ALIASING

It's worth touching on the subject of aliasing-a problem that plagues any sampled signal. The Nyquist sampling theorem states that a signal (image) may be completely reconstructed from a set of samples if the highest frequency it contains is less than twice the sampling frequency.

In simple terms, a sine wave must be sampled at least twice each cycle if it is to be reconstructed. Mathematically, this is often stated as $F_c \leq \frac{1}{2\Delta}$, where F_c is the maximum frequency component of the signal, and A is the sampling interval (in time or space).

Signals are sampled by taking measurements at regular intervals as illustrated in Figure 10a. The signal's time-domain function is multiplied by an infinite train of unit spikes called a Shah function.

Each spike takes on the magnitude of the signal at that point (see Figure 10b). But, the convolution theorem states that multiplication in one domain (e.g., time) is equal to convolution in the other domain (frequency, as shown in Figure 10c).

The FT of a train of spikes is just another train of spikes, and convolving a function with a train of spikes replicates it about the center of each spike—think of Figure 5 with a 1 in the center of the kernel surrounded by 0s. The result is an infinite plane of replicated FTs in frequency space, as you see in Figure 10d.

Now for aliasing. If the tails of the FTs die out before they touch, all is well. If they overlap, however, they add spurious frequencies to their neighbors, as you can see in Figure 10e. An alias of the tail results, seen as jaggies, moire, and/or ringing.

Aliasing can be avoided two ways. You can move the FTs farther apart in frequency space, which means moving the sample spacing closer. This means you're sampling at a higher frequency!

Or, you can limit the highest frequency of the original image so that its FT is no more than $\frac{1}{2\tau}$ wide (center to edge) and their tails then won't touch. Voilà, the Nyquist limit!

The best way to avoid aliasing is to filter out high frequencies before sampling, but this isn't always possible. Scans of half-toned pictures are a common example.

The easiest solution is to convolve the resulting sampled image with a gaussian to decrease the contribution of high frequencies in the FTs. This technique has the effect of preferentially reducing the strength of the aliasing signal, which is generally strongest out in the tails of the FT.

A better-but more expensive—solution is to sample at twice the final frequency (sample spacing) you plan to use. Convolve this with a gaussian to remove frequencies above F_c and then resample by interpolating down to the desired A.

A more specific technique is to identify the actual noise components that the signal has

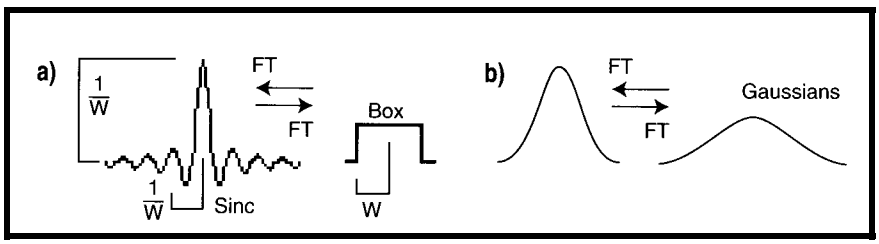


Figure 6—*These waves are Fourier transform pairs for a box and a gaussian. a—The box results in the complex sinusoid sinx/x. b—On the other hand, the FT of a gaussian results in just another gaussian.*

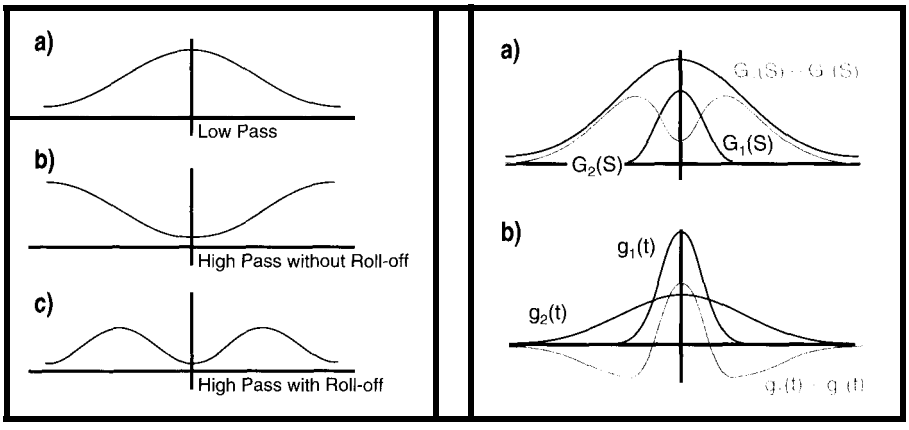


Figure 7—*Conceptually,* an image may be *filtered* by transforming *it to frequency space, tailoring its* frequency spectrum by multiplying it *by* a filter profile, *and then transforming the resulting* spectrum back *again. Shown here are the frequency profiles for three filters—low pass* ***(a),*** *high pass (**b**), and high pass with high-frequency roll-off (**c**).*

Figure **8a**—*High-pass fitters are conventionally constructed as the difference of two gaussians in* frequency space. These are shown in black, with *the difference in blue. b-These curves show the Fourier transform of this pair. Notice how the proportions of the two filter components reverse.*

been convolved with and then specifically deconvolve them out again.

TRANSFORM DETAILS

If you plan to do a lot of work, implement your own FFT and convolution codes. A good source is ***Numerical Recipes.*** **But,** if you just want to experiment on small images, you can go a long way with simple matrix algebra.

The Fourier (and related) transforms can be implemented as unitary transforms. These have the property:

$$T^{-1} = T^{*t}$$
$$T \bullet T^{*t} = I$$

where the Tare n x n matrices, • is matrix multiplication, * is the complex conjugate, and the prime represents the transpose. For real-valued unitary matrices:

$$T = T^{*t}$$

Two-dimensional image transforms are computed as the matrix products:

$$W = T \bullet w \bullet T^{*t}$$
$$w = T \bullet W \bullet T^{*t}$$

where w is an image, T is the matrix containing the transform's coefficients, and ***W*** is the transform of w. Notice the reciprocal symmetry between w and W-they are indistinguishable in the equations.

As an experiment, you can generate the transform's basis images by inverse transforming coefficient matrices containing only one 1 (with the rest as 0s). Let $D^{i,j}$ be a matrix that is all 0s except for element (i, j). Then, the corresponding basis image $B^{i,j}$ is:

$$B^{i,j} = T \bullet D^{i,j} \bullet T$$

There are n^2 of these, each corresponding to a different location for 1. The 64 components of Figure 2 were generated in this way.

Finally, you need to generate transformation matrices, which must be the same size (rows and columns) as the image matrices they will be applied to. The elements W_{ij} of the general 2D transform matrix are:

$$\begin{vmatrix} W_{00} & \cdots & W_{0,n-1} \\ & \ddots & \\ W_{n-1,0} & \cdots & W_{n-1,n-1} \end{vmatrix}$$

For the 2D DFT, these elements are defined by:

$$W_{i,k} = \frac{e^{-j2\pi\frac{ik}{n}}}{\sqrt{n}}$$

where n is the transform's size (i.e., its row width), i,k is the row, column location of the element (0 to $n-1$), j is the square root of –1, and e is 2.71828.... Remember, you can move factors that multiply every element of a matrix out in front of the matrix.

The discrete cosine transform avoids complex numbers and is widely used for compression (e.g., JPEG):

$$C_{i,k} = \frac{\alpha_k}{\sqrt{n}} \cos\left[\frac{\pi(2i+1)k}{2n}\right]$$

$$\alpha_k = \begin{cases} 1, m = 0 \\ \sqrt{2}, m > 0 \end{cases}$$

The Hadamard transform features all integer arithmetic (keep the square root of 2 factors outside of the matrix). The basic Hadamard transform starts with this 2 x 2 matrix:

$$H_2 = \frac{1}{\sqrt{2}} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

These are then recursively embedded into the following block matrix to generate successively larger transforms:

$$H_n = \frac{1}{\sqrt{2}} \times \begin{bmatrix} H_{\frac{n}{2}} & H_{\frac{n}{2}} \\ H_{\frac{n}{2}} & -H_{\frac{n}{2}} \end{bmatrix}$$

IMAGE TRANSFORMATION

Images have frequency spectra just like any other signal. These spectra may be manipulated by transforming an image to frequency space and tailor-

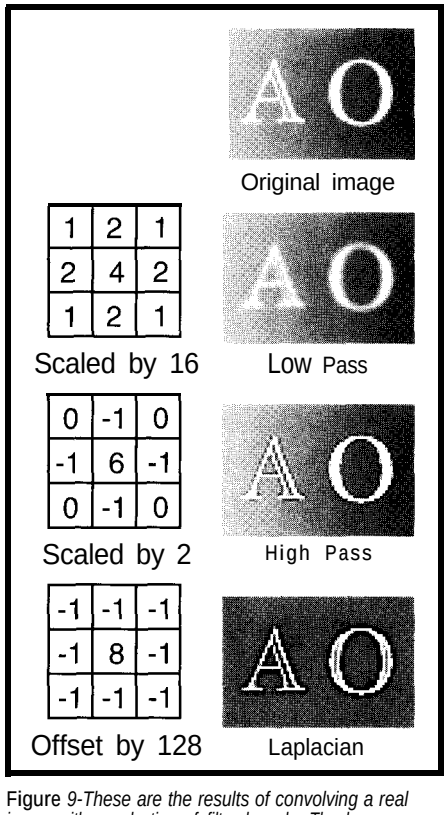


Figure 9-*These are the results of convolving a real image with a selection of filter kernels. The low-pass kernel has a gaussian profile. High pass uses the profile developed in Figure 8b. A laplacian image operator results when the weights sum to 0. The result must be offset by 128 to make it visible.*

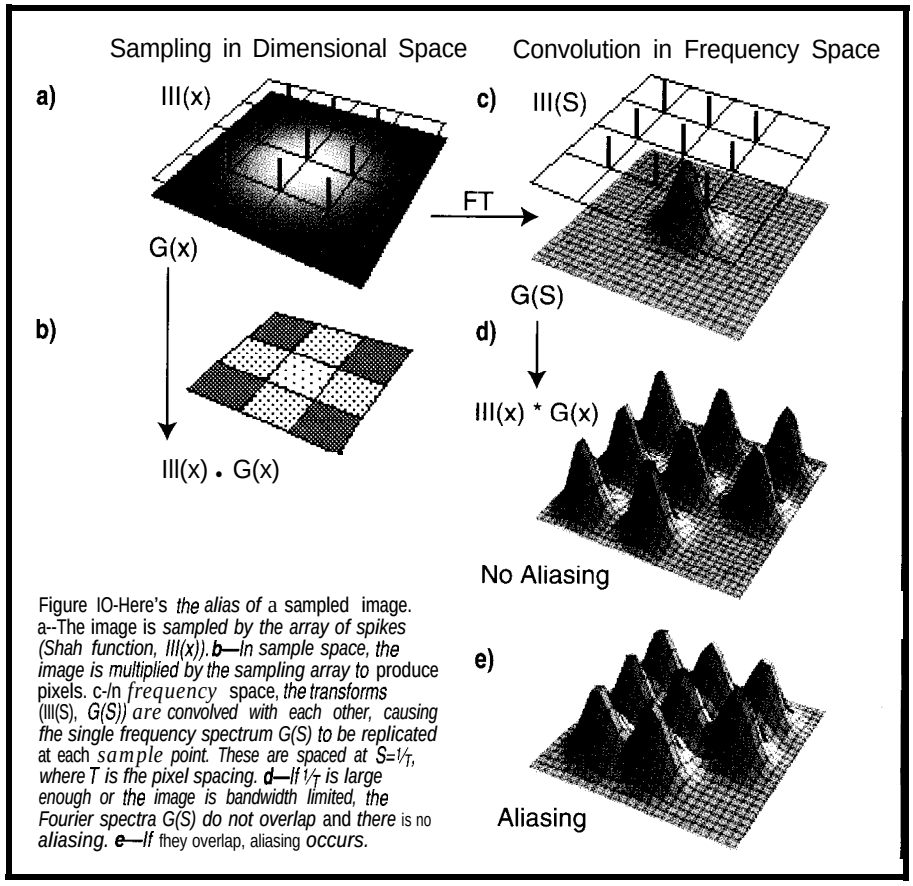


Figure IO-Here's *the alias of* a sampled image. a--The image is *sampled by the array of spikes (Shah function, III(x)). b—In sample space, the image is multiplied by the sampling array to* produce pixels. c-/n *frequency* space, *the transforms* (III(S), *G(S)) are convolved with each other, causing fhe single frequency spectrum G(S) to be replicated* at each *sample point. These are spaced at S=1/T, where T is fhe pixel spacing. d—If 1/T is large enough or the image is bandwidth limited, the Fourier spectra G(S) do not overlap* and *there* is no *aliasing. e—If* fhey overlap, aliasing *occurs.*

ing its frequency profile by simply multiplying the spectrum with the desired filter profile. The resulting spectrum is inverse transformed back into a real image.

This process provides a conceptually simple basis for low-pass (blurring) and high-pass (sharpening) filtration. Operationally, it's often easier and faster to use the equivalent process of convolution to get the same results.

However a filter is implemented, it's important to be aware of how narrow- and wide-frequency spectra transform. The most important consequence is aliasing, which results when overly broad frequency spectra overlap, often due to undersampling and / or poor filter design.

To learn more, take a look at efficient filter implementation algorithms, sampling theory, directional filters, the cosine transform, deconvolution, and wavelet transforms. Good hunting. ❑

Many thanks to Mathematica for Figures 1–4, 8, and 10 and to Photoshop for Figure 9.

Bruce Hubbard has worked on imaging processes at TRW, JPL, and COMTAL/3M. His interests include expert systems, and he is currently president of Direct Imagination, a producer of educational multimedia.

REFERENCES

K.R. Castleman, *Digital Image Processing,* Prentice-Hall, Englewood Cliffs, NJ, 1996.

A. Glassner, *Principles of Digital Image Synthesis,* Vol. 1, Morgan Kauffman Publishers, San Francisco, CA, 1995.

W.H. Press et al., *Numerical Recipes in C,* Cambridge University Press, New York, NY, 1992.

SOURCES

For Web resources and IP shareware, check out the HREM facility at Northwestern University at <http://risc1.numis.nwu.edu/internet/>.

I R S

407 Very Useful
408 Moderately Useful
409 Not Useful

The Global Positioning System

Part 1: Guiding Stars

Sure, you can buy a GPS receiver for less than $200, but you also need to understand them. Do-While explains positioning equations, clock accuracy, receiver design, selective availability, and differential correction.

FEATURE ARTICLE

Do-While Jones

There's a constellation of 24 man-made stars circling our planet, radiating precise position information. Fishermen use them to return to their favorite fishing holes. Pilots find airports in bad weather with them.

These stars tell surveyors where property lines are. They enable geologists to measure movements in the earth's crust that are only a few centimeters per year. And, information from these stars can drive a moving map display that shows you where you are and how to get to your destination.

For as little as $200, you can purchase a receiver like the one in Photo 1 that tells you exactly where you are when you're hiking in the woods.

But since you're reading *INK*, it's a good bet that, at some point in time, you may be asked to build a receiver yourself or integrate it into something you designed. At the very least, you're probably curious about how the Global Positioning System (GPS) works.

BASIC EQUATIONS

If you ignore all the details, it's pretty simple. So, let's start out by ignoring all those annoying gnats. Once we've established the basics, we can examine the problems and how to solve them.

Suppose you are at some unknown point P with coordinates (x, y, z). You discover that you are some distance R_1 from point P_1 with coordinates (x_1, y_1, z_1).

You know from the 3D version of the Pythagorean Theorem that:

$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = R_1^2$$

Then, if you discover you are distance R_2 from point P_2 and distance R_3 from point P_3, you have two more equations:

$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = R_2^2$$
$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = R_3^2$$

You have three equations with three unknowns, so the equations can be solved.

Since these are quadratic equations, there are two outcomes because of a plus-or-minus square-root term in the solution. One solution will be on or near the surface of the earth. The other will be about 20,000 miles above its surface. I think you can figure out which is correct.

In theory, if you know your distance from three GPS satellites, you can determine where you are. But, how do you know this distance? You calculate it from the time it takes for a radio signal from the satellite to reach you.

Every satellite has a very accurate clock (more about that later) that broadcasts signals at precise times. All you do is measure the precise time you receive them.

CLOCK ACCURACY

The first problem is clock accuracy. Radio waves travel ~1′ per nanosecond, so if your clock is 1 µs fast or slow, your distance measurements can be off by 1000′.

You have two options. Option 1—use a very precise, very expensive clock. Option 2—figure out a way to compensate for your clock's errors. Option 2 wins, hands down.

As it turns out, it is very easy to compensate for receiver clock errors. The distance to point P_1 is computed by:

$$R_1 = c \times t_1$$

where c is the speed of light and t_1 is the measured time for the radio signal from the satellite at P_1 to reach the receiver.

But, the true distance is really:

$$R_1 = c\,(t_1 + t_e)$$

where t_e is the time error of the re-

ceiver clock. Similarly, the true distance to point P_2 is:

$$R_2 = c\,(t, + t,)$$

where t_e is the same time error of the receiver clock.

So, you really have three equations with four unknowns. The simple solution-measure the distance to a fourth satellite-gives you four equations with four unknowns (x, y, z, and t_e).

As long as the clock doesn't drift appreciably while you measure the four distances, you can remove all the receiver clock errors. Since you can remove the errors, you can use GPS to build a clock that's accurate to ±150 ns.

In the Navstar system shown in Photo 2, 24 GPS satellites are arranged so that at least four are always in view. Since six satellites are usually visible, you can overconstrain the solution and improve the accuracy of your position with the redundant data.

SATELLITE SYNCHRONIZATION

Of course, the satellite clocks have to know the right time, too. The U.S. government spent big bucks for some of the best clocks money can buy and put them in these satellites.

But, even these clocks need to be corrected from time to time. They are stable enough that they drift so little in a 12-h period that setting them twice a day is more than adequate.

The tax dollars spent on these satellites came from the Department of Defense's budget. These satellites are up there to help American military forces navigate to precise locations.

Most places the U.S. government has traditionally wanted to destroy are outside the U.S. If the U.S. had parked navigation satellites in geosynchronous orbit over foreign countries, it's pretty unlikely that the leaders of those countries would be willing to set the satellite clocks every day just so the U.S. could blast them to bits any time they desired.

Therefore, the satellites are in orbits that pass over the U.S. every **12** h. This gives U.S. tracking stations an opportunity to accurately measure the trajectories of the satellites and correct their orbits and clocks.

How accurate are these clocks? GPS satellites use 10.22999999543.MHz oscillators to compensate for relativistic effects. To observers on the ground, it appears that they are running at exactly 10.23 MHz. Yes, this truly is rocket science.

SATELLITE POSITION

The fact that the satellites are moving is another one of those nasty little details. The positions of points P_1, P_2, P_3, and P_4 are all functions of time. So, the GPS receiver needs to know where the satellites are at any given time.

When the ground stations correct the satellites' clocks, the ground stations also send them parameters that describe their current orbits. The satellites broadcast these parameters to the GPS receivers in the precisely timed messages they transmit.

Getting a GPS reading is not trivial. Timing errors, satellite positioning, weak transmission signals.. . all add up to fancy engineering challenges.

When the GPS receiver gets the message, it must do three things. It must determine the exact time it received the beginning of the message. It must decode the message's time tag to determine when it was sent. It must read the orbital parameters and compute the satellite's position at the time the message was sent.

RECEIVER DESIGN

Suppose all 24 satellites transmitted their information on separate frequencies. You'd have to build at least four tunable receivers to receive data on the four different frequencies. That could be expensive.

To maintain lower costs, all the satellites broadcast simultaneously on the same frequency. You might think this would cause terrible interference.

It doesn't really. The signals are so weak that by the time they reach the GPS receiver they're already 65 dB below the noise floor! They're so weak, they won't cause interference.

There are some good reasons why the signals are so weak. The satellites have limited power and are in very high orbit.

As well, GPS receivers usually have size and weight limitations (e.g., they have to fit in a missile or be carried by a person) which prevent them from having big, high-gain antennas. Even if you use a narrow-beam, high-gain antenna, you can't aim it at four satellites at once.

Of course, receiving -65 dB signals isn't trivial, but it's possible if you trade time and bandwidth for power. Each satellite transmits a different, known pseudonoise (PN) sequence. The receiver on the ground correlates the received signal with a reference PN sequence that's identical to the one used by the satellite.

If the transmitted signal's polarity matches the polarity of the reference sequence, the correlator output is positive. If the polarity of the transmitted signal is reversed, the output is negative. And, if the transmitted signal doesn't match the phase and bit pattern of the reference sequence, the average output is zero.

The receiver adjusts the phase of its reference sequence until it correlates with a satellite sequence. When this happens, the correlator gives a strong positive or negative output.

The satellite uses the polarity of its PN sequence to transmit 1s and 0s. The correlator output yields this binary series, allowing the receiver to decode the data. The data includes the time when the frame started and the satellite's orbital parameters.

The amount of phase shift necessary to make the sequence correlate tells the time it took the signal to get from the satellite to the receiver, which is proportional to the distance. The receiver uses early and late correlation gates to stay phase locked with the transmitted PN signal until the satellite disappears over the horizon.

It isn't quite this simple. Since the satellites move fast enough to circle the earth twice per day at a very high orbit, there's a Doppler shift on the

satellite's signal. The frequency is higher than nominal as the satellite rises above the horizon and lower than nominal as it sets.

Since the receiver is trying to receive a –65-dB signal, the receiver IF (intermediate frequency) bandwidth has to be as small as possible to cut out the noise. The Doppler shift is larger than the IF bandwidth, so you have to tune the receiver to account for the Doppler shift.

Photo 1—*GPS technology has advanced to the point that hand-held receivers like this one are affordable for recreational use.*

WARM-UP TIME

A GPS receiver contains a microcomputer that remembers each satellite's PN code and its last set of orbital parameters. The microcomputer is programmed to predict when the various satellites rise and set.

It can therefore optimize the search for new satellites by looking at the right frequency for the correct code and phase shift. It can find the satellite shortly after it comes into view.

But, when you first turn on a GPS receiver, it doesn't have any idea what satellites are on its side of the earth. It doesn't know if the satellites are rising or setting. So, it has to guess many combinations of satellite codes, code phases, and Doppler shifts before it can dig a satellite's signal out of the noise—and this can take a *long* time.

This so-called warm-up time is similar to the delay in old tube-radio receivers. Remember how you had to wait a while after you turned on the radio before you could hear any music? It took time for the tubes' iilaments to warm up enough for the electrons to flow through the vacuum tube. The GPS receiver's delay is even longer.

The classic way the warm-up problem was solved for vacuum tube radios was to leave the radio on all the time. The volume was turned all the way down when it wasn't being listened to.

You should do the same thing with a GPS receiver. Once it's on, don't turn it off if you can help it!

If you design a GPS navigator system for a car, don't turn off the receiver when the ignition is switched off. Turn off the display, but keep the receiver, correlators, and microprocessor running. Otherwise, the car probably won't know where it is until after it reaches its destination.

If you carry a hand-held GPS receiver into a metal building, it's going to lose all the satellite signals. If it's smart and has a stable clock, it can dead-reckon the satellites' positions. And, if you don't spend too much time inside, it can acquire the satellites quickly when you emerge.

But, if you leave the receiver someplace where it can't receive signals for several months, don't expect it to tell you where you are when you take it out of storage.

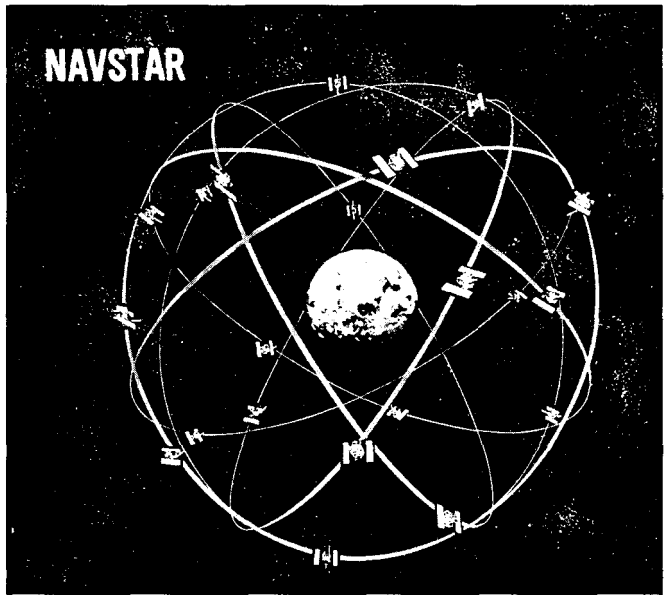


Photo 2-A *constellation of 24 navigation satellites circles the globe twice per day. Data broadcast continuously from these satellites can be used to find your position.*

MULTIPLE CHANNELS

The receiver needs to know the distance to at least four satellites to compute its position. Although the satellites all transmit the same frequency, the Doppler shift makes them appear at slightly different frequencies.

The preamplifier probably has a wide enough bandwidth to pass all the satellite signals. But, the IF stage needs at least one demodulator consisting of a tunable IF filter and correlator.

A single demodulator can be multiplexed to find the distances to the four satellites sequentially. This makes processing more difficult because the distances to the four satellites are found at four different times. It has to extrapolate the data to find the distances at a common time.

If the position of the receiver moves during the time it takes to make the four distance measurements, accuracy suffers. Better receivers have at least four demodulators (i.e., channels), so they can determine the position to four satellites at once. A six-channel receiver can track six satellites simultaneously.

SELECTIVE AVAILABILITY

The U.S. didn't put all these satellites in space just so your golf cart can tell you how far the next green is—although it can, as you see in Photo 3! GPS was developed by the U.S. military for military purposes.

But, if the U.S. can use GPS satellites to guide its weapons to precise locations in foreign countries, then foreign countries can use them to attack U.S. targets. Since this concerns the Department of Defense, they tried to solve the problem by encrypting satellite signals.

Signals from the satellites are no good unless you know when they were sent and where the satellites are. If the transmission time and orbital parameters broadcast by the satellites are encrypted, the satellites are useless to any receiver without the encryption key.

Such action would mean that enemy weapons couldn't use GPS to attack American targets. But, it also would mean commercial airliners couldn't use GPS to find airports, hikers couldn't use GPS to find their way through the woods, surveyors couldn't.. .

Since it would be a shame to have GPS technology in place but unusable to anyone except the military, the military graciously added a second, lower accuracy channel for commercial use.

This channel is not encrypted, and it is used by all the commercial GPS products that are becoming so popular. This secondary channel has lower resolution, and the military intentionally introduces errors to degrade the accuracy.

The politically correct term for these little white lies is Selective Availability (SA). SA errors are unacceptable for GPS receivers of commercial aircraft autopilots. The airplane must land gently on the center line of the runway. So, the Federal Aviation Administration uses differential correction to give back what the DOD takes away.

DIFFERENTIAL CORRECTION

If you have a GPS receiver on the ground at a known location, you can compare the true location with the slightly incorrect location you get from the GPS receiver. You then broadcast differential corrections that allow other GPS receivers to figure out where they are by removing the errors intentionally inserted by the military on the civilian channel.

Of course, if American companies can build autopilots that use differential correction to land airplanes precisely on the runways at Chicago's

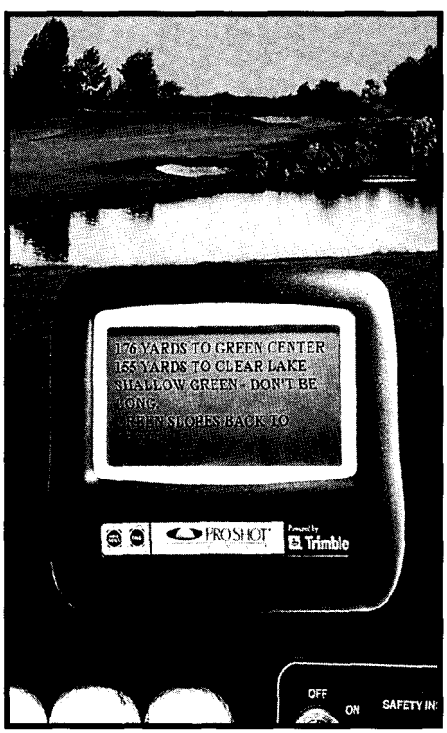


Photo 3-A *GPS receiver can be* combined with a database containing positional data to create a product *that does more than simply tell you your position.* This golf **cart** can *tell* you *how* far you *are from the next green.*

O'Hare Airport, then foreign countries can use differential correction to build an autopilot that sends a missile through the window of any specified building in Chicago.

If U.S. defense systems detect a weapon flying toward Chicago, they could increase SA errors and make the FAA shut off their differential-correction broadcasts. The weapon might miss the intended target-and hit an elementary school instead! It might also guide Flight 1256 through the fog and straight into Lake Michigan. Most military experts consider that to be an inappropriate countermeasure.

The civilized world got along for thousands of years without telephones and computers. Now, we can't function without them. Our increasing reliance on GPS may someday make it just as essential. It may soon be just as impractical to turn off (or seriously degrade the data from) the GPS satellites as it would be to shut off all our computers and phones.

Besides, it's clear that one doesn't need GPS to navigate a truck full of explosives into the World Trade Center parking garage or to the Federal Building in Oklahoma City. If anyone *did* attack Chicago (with or without GPS-guided weapons), they could expect a retaliatory strike of such magnitude that they'd surely regret the attack.

That's why there's a growing sentiment that SA is unnecessary, ineffective, and just plain stupid. The military is considering the proposal to stop intentionally introducing errors in the civilian channel. I don't expect unrestricted access to uncorrupted GPS data for a few more years, but sooner or later, I think it has to happen.

OTHER ACCURACY FACTORS

There are two other problems to address-refraction and multipath.

Radio waves tend to bend as they pass through the atmosphere, especially at low elevation angles. Military systems correct for this effect using both classified and unclassified data.

These two channels operate in different frequency bands which are affected differently by refraction. You can determine the amount of refraction by determining the difference between the two signals' times of arrival.

If there is no difference, then refraction is negligible. If the difference is large, you can compensate for the large refraction.

Multipath causes trouble two ways. First, destructive interference can make the signal disappear. Second, the indirect path can make the satellite appear farther away than it actually is, resulting in an incorrect position computation. Multipath can be minimized by proper antenna placement and clever antenna design.

Geologists can get highly accurate measurements of earth crust move-

ments because they can be very patient. Time can buy more accuracy. They can set their GPS receivers on the rocks of interest and take data every few minutes for a couple of days. The errors introduced by SA have zero mean. After averaging all those measurements, geologists come pretty close to the true position.

Once they are close, a clever trick gets them even closer. Although they can't decipher the encrypted data sent on the military channel, they can tell when the bits change state.

After days of observations, they know approximately when each message must start. If they know that time to within half the bit period of the military PN code, they can figure out which bit transition marks the beginning of the encoded military PN sequence.

Once the start bit is identified, they can determine the precise time when the military PN code begins, even though they still can't read it.

Then, they can take more measurements using the start times of the (indecipherable) military PN code to further refine their position measurement.

That's how geologists get measurements accurate enough to measure motion that is only a few centimeters per year.

USING GPS

GPS can tell you where you are in geocentric coordinates. But unless you're flying a spacecraft, you probably don't want to know how far you are from the center of the Earth.

You probably want to know your latitude, longitude, and altitude so you can find yourself on a map. You may want to know how to use GPS to navigate from one location on the surface of the earth to another.

Next month, I'll show you how to take the x, y, and z GPS positions and convert them into useful working coordinates. ❑

Many thanks to Lea Ann McNabb of Trimble Navigation for providing photos **for** this article.

Do- While [ones has been employed in the defense industry since 1971. He has published more than 50 articles in a variety of popular computer magazines and has authored the book Ada in Action. You *may reach him at do_while@ridgecrest.ca.us.*

SOURCES

GPS products and services

Navtech GPS Supply
2775 S. Quincy St., Ste. 610
Arlington, VA 22206
(703) 931-0500
Fax: (703) 93 l-0503

Trimble Navigation Ltd.
645 N. Mary Ave.
Sunnyvale, CA 94086
(408) 481-8000
Fax: (408) 481-7744

I R S

410 Very Useful
411 Moderately Useful
412 Not Useful

DEPARTMENTS

Applying the HCS II

Part 1: Of Consoles and Criminals

The joy of home automation is getting the system to give you what you need—whether it's automatic lighting or simulated activity for those times you are away. Ken offers some tips from his own HCS II setup.

MICRO SERIES

Ken Davidson

So, what are you doing with your HCS II? That's the most common question I've heard since the HCS II was first introduced almost five years ago.

Since everyone's situation is unique, I've always been reluctant to provide sample code. I can't anticipate what code would be useful.

As well, sample code might artificially restrict what people think can be done. It's better for you to decide how you want it to work and then worry about the implementation.

Steve and I have both been asked to publish the code we use for our houses, so people get ideas of what can be done. But, that's like asking me to publish my will. Sure, it might help you, but it gives strangers an unwelcome glimpse into my private life.

Instead, Steve and I are giving you a peek into what we're using in our own HCS II installations without telling all. My examples might be obvious to some but will offer others a springboard for their own projects.

Next month, Steve will introduce a new HCS network module based on the Answer MAN serial data-acquisition module. He'll show how he's using several of them in a detached building to replace unreliable X-10 modules.

USER INPUT

I'm often asked what I use as a console to interact with my HCS II setup. My quick answer is, nothing. Instead, I rely primarily on passive input to the system, and the system reacts with preplanned events.

For example, if the light-level sensor shows that it's dark out and the

Listing 1-hexpensive *X-10 minicontrollers* can be used to send commands to *the HCS.*

```
DEFINE Mode             = Variable(.0)
DEFINE Daytime          = 1
DEFINE Evening          = 2
DEFINE Night            = 3
DEFINE Movement         = Variable(20)
DEFINE NightOn          = Module(H1
DEFINE Chandelier       = Module(L1
DEFINE Torchiere        = Module(L2
DEFINE EndLight         = Module(L3
DEFINE KitchenLight     = Module(L5
DEFINE DiningLight      = Module(L6
DEFINE BathroomLight    = Module(L9
DEFINE HallLight        = Module(L1 )
DEFINE BedLight         = Module(B1
DEFINE KidsLights       = Module(B9
DEFINE HallMotion       = Input 183)
DEFINE BathroomMotion   = Input 182)
DEFINE LivingMotion     = Input 181)
DEFINE KitchenMotion    = Input 180)
DEFINE MasterMotion     = Input 179)
DEFINE KidsMotion       = Input 178)
DEFINE KidsAwake        = Input 161)
DEFINE HallTimer        = Timer 4)
DEFINE BathroomTimer    = Timer 67)
DEFINE KidsTimer        = Timer 68)
DEFINE KidsTimer2       = Timer 5)
DEFINE LivingTimer      = Timer 69)
DEFINE KitchenTimer     = Timer 70)
DEFINE MasterTimer      = Timer 71)
DEFINE HouseMotion      = Timer 73)
DEFINE RandomTimer      = Timer 74)
DEFINE HallDimmed       = Variable(12
DEFINE BathDimmed       = Variable(13
DEFINE LightLevel       = ADC(128)
DEFINE DayEveningLow    = 55
DEFINE DayEveningHigh = 70
!=============================================================
BEGIN

IF Reset THEN
  Refresh = 0; Mode = Daytime
END
!-------------------------------------------------------------
IF LightLevel<=DayEveningLow THEN
  Mode = Evening
END
IF LightLevel>DayEveningHigh THEN
  Mode = Daytime
END
IF LightLevel<=DayEveningLow AND Mode=Night AND Time>6:00:DY AND
 Time<20:00:DY THEN
  Mode = Evening; HallLight = OFF; BathroomLight = OFF
  HallDimmed = FALSE; BathDimmed = FALSE
END

IF NightOn=ON THEN
  Mode = Night: NightOn = OFF
END
!-------------------------------------------------------------
IF Mode=Night THEN
  BedLight=OFF; BedTVTimer=ON; Chandelier=OFF: EndLight=OFF;
  Torchiere=OFF; KitchenLight=OFF; DiningLight=OFF; Sin kLight=OFF;
  DeckLight=OFF;  LowerLanding=OFF:  GarageInside=OFF;
  GarageOutside=OFF; FrontDoor=OFF; BasementLights=OFF;
  HallLight=Dim(20); BathroomLight=Dim(20);
  HallDimmed=TRUE;  BathDimmed=TRUE;
END
!-------------------------------------------------------------
IFA Mode=Night THEN
  IF HallMotion=ON AND HallDimmed=TRUE THEN
    HallLight = Bright(8); HallDimmed = FALSE
  END
END

IF HallMotion=ON THEN
  HallTimer = ON
END
IFA HallTimer>30 THEN            ! 30 s
  IFA Mode=Night THEN
    HallLight = Dim(20); HallDimmed = TRUE: HallTimer = OFF
```

(continued)

motion sensor sees movement in the living room and if we're not in bed yet, the living-room lights come on. I didn't actively do anything, but the system still received input from me.

However, there are still times when it's nice to "tell" the system to go into a different state. It doesn't require a full-blown console, nor does it even require an LCD display and push buttons. All you need are unobtrusive devices you already have in the house.

A moment ago, you may have asked, "So, how does the system know he's in bed?" I use a simple trick that takes advantage of the HCS's two-way X-10 capability and the X-10 minicontrollers I already have around the house.

Listing 1 shows some sample code. I assign a dummy house code and module number [e.g., H1) to an event I want to take place (e.g., bedtime). When we climb into bed, I reach over to the minicontroller next to the bed and press the "H1 On" button.

When the HCS sees module H1 change state, it initiates a series of actions. It turns off all the lights in the house, dims the hall and bathroom lights all the way down, and goes into night mode.

There is no physical H1 module, but the HCS doesn't know that. It simply receives the On command and acts on it. Notably, the last step in the action list is to turn module H1 off. That way, the HCS can sense when I press the On button again.

As an aside, let me point out another little trick that gets around a shortcoming of X-10 lamp modules. As I'm sure you've found out the hard way, when a lamp module is off, it must come full on before it can be dimmed. That may be fine in the evening when you're adjusting the lights for watching TV, but it's rather rude to be blinded in the middle of the night.

However, a lamp module that's dimmed all the way to black can be gradually brightened in a soft, controlled manner. As part of the good-night routines, the HCS dims the hall and bathroom lights to black.

The lights look like they're off and they don't use any power, but the lamp module still considers them on. When someone gets up during the night, the

HCS can brighten the hall and bathroom lights just enough to light the way, without blinding the person.

When the sun comes up and the HCS automatically goes into day mode, it sends true Off commands to the lights so they are ready to blink full on in the evening when it gets dark out.

NAP TIME

I have motion detectors in every room of the house. Combined with the light-level sensor and evening and night modes, the HCS can pretty effectively know when the lights are supposed to be on or off in specific rooms.

It worked fine until we threw a baby into the picture. With babies come naps.

When our first daughter was born, I set the system up so the lights in her room wouldn't come on at all during the day-even on dark, rainy days. That way, they never came on during naps.

I also picked an arbitrary time in the evening when the system stopped turning on the lights in her room in response to motion. That way, we could check her after she went to bed without disturbing her.

But, add a second baby to the same room with different naps and bedtimes. I needed an easy way to tell the system that someone was sleeping. I could have used an X- 10 minicontroller like I do at our bedtime, but I wanted it to be even easier to use and instantly apparent when it was in nap mode.

The light switch turned out to be the perfect solution. I didn't want to rewire the room, so I plugged a wall transformer into the outlet that's controlled by the switch.

I ran the low-voltage output to a direct input on the HCS. When one of the girls goes to bed, we simply flip off the wall switch and the HCS no longer responds to motion in the room. When they wake up, we flip the switch on (see Listing 2).

How does this differ from simply manually controlling the room lights? Flipping the light switch on doesn't necessarily cause the lights to come on. They only come on if it's dark enough outside to require them. And, the lights still come on automatically in the evening when the girls go to bed.

Listing 1-continued

```
  ELSE
    HallLight = OFF: HallTimer = OFF
  END
END
!-----------------------------------------------------------------
IFA Mode<>Night THEN
  IF BathroomMotion=ON THEN
    BathroomLight = ONA; BathroomTimer = ON
  END
  IF BathroomTimer>8 THEN
    BathroomLight = OFF: BathroomTimer = OFF
  END
END

IFA Mode=Night THEN
  IF BathroomMotion=ON THEN
    BathroomTimer = ON
  END
  IF BathroomMotion=ON AND BathDimmed=TRUE THEN
    BathroomLight = Bright(8); BathDimmed = FALSE
  END
  IF BathroomTimer>3 THEN
    BathroomLight=Dim(20); BathDimmed=TRUE; BathroomTimer=OFF
  END
END
```

I added one more function to the setup. Once in a while, we get the girls up without opening the shades. Since it's light out, the HCS won't turn the room lights on. But, since the shade is closed, we may want them on anyway.

As the latter half of Listing 2 shows, if I flip the switch off and on within 2 s, the system turns the lights on regardless of the outside light level.

LIVED-IN LOOK

Experts agree that one way to dissuade burglars is to make your house look as lived-in as possible when you're away. Simple timers are a start, but burglars can see the regular pattern if they watch your house for a few days. The HCS and its smarts can improve that lived-in look.

The technique I use separates the steps to turn lights on and off (see Listing 3). I first deal with motion (or simulated motion) in a room. Then, I decide when to turn lights on and when to turn them off.

Obviously, when someone is home, motion detectors decide if someone is in a room. That input, the time of day, outside light level, and so on determine when a light should come on.

A fixed timer decides when to turn lights off. By decoupling the turn-off control from the turn-on control, I can have the system turn off lights that were turned on manually.

Listing *2-Another unobtrusive input* device is a *standard wall switch controlling a low-voltage transformer which then goes to an input on the HCS.*

```
IFA Mode=Evening THEN
  IF KidsMotion=ON AND KidsAwake=ON THEN
    KidsLights = ON
  END
END

IF KidsMotion=ON THEN
  KidsTimer = ON
END
IF KidsTimer>5 OR KidsAwake=OFF THEN
  KidsLights = OFF
  KidsTimer = OFF
END

IF KidsAwake=OFF THEN
  KidsTimer2 = ON
END
IF KidsAwake=ON AND KidsTimer2<2 THEN
  KidsLights = ON; KidsTimer = ON; KidsTimer2 = OFF
END
```

Listing 3-For *that* lived-in look when you're away, the *HCS simulates motion throughout the house.*

```
IF Reset THEN
  Refresh = 0; Mode = Daytime; HouseMotion = ON
END
!-------------------------------------------------------------------
IF HouseMotion>60 THEN
  Away = TRUE
ELSE
  Away = FALSE
END
!-------------------------------------------------------
IF Away=TRUE AND Mode<>Night THEN
  RandomTimer = ON
ELSE
  RandomTimer = OFF
END

IF RandomTimer>=10 THEN
  RandomTimer = ON: Movement = Random(200)
  IFA Movement>=100 AND Movement<150 THEN
    LivingMovement = TRUE
  END
  IFA Movement>=150 AND Movement<165 THEN
    KitchenMovement = TRUE
  END
  IFA Movement>=165 AND Movement<170 THEN
    BathroomMovement = TRUE
  END
  IFA Movement>=170 AND Movement<185 THEN
    HallMovement = TRUE
  END
  IFA Movement>=185 AND Movement<195 THEN
    KidsMovement = TRUE
  END
  IFA Movement>=195 AND Movement<200 THEN
    MasterMovement = TRUE
  END
END

IF LivingMotion=ON    THEN LivingMovement = TRUE    END
IF KitchenMotion=ON   THEN KitchenMovement = TRUE   END
IF BathroomMotion=ON  THEN BathroomMovement = TRUE  END
IF HallMotion=ON      THEN HallMovement = TRUE      END
IF KidsMotion=ON      THEN KidsMovement = TRUE      END
IF MasterMotion=ON    THEN MasterMovement = TRUE    END
!-------------------------------------------------
IFA Mode=Evening THEN
   IF LivingMovement=TRUE THEN Torchiere = ON: EndLight = ON END
   IF HallMovement=TRUE THEN HallLight = ON END
   IF MasterMovement=TRUE THEN BedLight = ON END
   IF KidsMovement=TRUE AND KidsAwake=ON THEN KidsLights = ON END
   IF KitchenMovement=TRUE THEN KitchenLight = ON END
END

IF LivingMovement=TRUE THEN
  LivingTimer = ON; HouseMotion = ON LivingMovement = FALSE
END
IF LivingTimer>=20 THEN
  Torchiere = OFF; EndLight = OFF: LivingTimer = OFF
END

IF KitchenMovement=TRUE THEN
  KitchenTimer = ON; HouseMotion = ON: KitchenMovement = FALSE
END
IF KitchenTimer>=5 THEN
  KitchenLight = OFF; DiningLight = OFF; KitchenTimer = OFF
END

IF HallMovement=TRUE THEN
  HallTimer = ON: HouseMotion = ON: HallMovement = FALSE
END
IF HallTimer>=30 THEN
  HallLight = OFF; HallTimer = OFF
END

IF MasterMovement=TRUE THEN
  MasterTimer = ON: HouseMotion = ON; MasterMovement = FALSE
END
IF MasterTimer>=10 THEN
  BedLight = OFF; MasterTimer = OFF
END
```

(continued)

Listing 3—*continued*

```
IF KidsAwake=OFF THEN
  KidsTimer2 = ON
END
IF KidsAwake=ON AND KidsTimer2<2 THEN
  KidsLights = ON
  KidsTimer = ON; KidsTimer2 = OFF
END
IF KidsMovement=TRUE THEN
  KidsTimer = ON; HouseMotion = ON
  KidsMovement = FALSE
END
IF KidsTimer>5 OR KidsAwake=OFF THEN
  KidsLights = OFF
  KidsTimer = OFF
END
```

The only difference between a vacant house and an occupied house is the lack of input to motion detectors. All other factors should remain the same. If you want a truly lived-in look, you need to simulate motion.

I selected six areas of the house that we spend the vast majority of our evenings in and that contain lights visible from outside. I determined the percentage of time we spend in each of those areas. Table 1 lists those numbers.

When the HCS hasn't detected any motion in the house for an hour, the system goes into away mode. It generates a random number between 0 and 200 every 10 min. and uses that number to simulate motion in the house.

I figure that 50% of the time there's no motion at all. That covers values ranging O-100. For the other half of the range, I use the numbers in Table 1.

So, when the random number falls between 100 and 150, for example, the HCS sees "motion" in the living room. The code handles that motion as if there were a person in that room.

It turns on the appropriate lights, starts the timer for that area, and turns the lights off when the timer runs out. If simulated motion is detected in the same room before the timer times out, it retriggers the timer just as someone moving in the room would do.

Note that it's still possible to react to real motion in a room if the system is set up to generate an alarm condition when the homeowner is away. Simulated motion simply turns a light on. Real motion triggers the sirens.

I'm still fine-tuning the time between random numbers and percentage values assigned to each area, but it certainly doesn't have to be perfect. And, it definitely beats simple timers.

Area	Chance of Movement
Living Room	50%
Kitchen	15%
Bathroom	5%
Hallway	15%
Kids' Room	10%
Master Bedroom	5%

Table I-Motion is simulated throughout the house by *assigning values to each room fhaf represent fhe likelihood of motion.*

JUST THE TIP

I've barely scratched the surface of what I'm doing with the system, so perhaps I'll write up some more examples in the future.

Next month, Steve takes the helm and spends time with some new hardware and the XPRESS code to run it. ❑

Ken Davidson is the editor-in-chief and a member of Circuit Cellar INK's *engineering staff. He holds an M.S. in computer engineering from Rensselaer Polytechnic Institute. He may be reached at ken.davidson@circellar.com.*

SOURCE

HCS II
Circuit Cellar, Inc.
4 Park St.
Vernon, CT 06066
(860) 8752751
Fax: (860) 871-0411

I R S

413 Very Useful
414 Moderately Useful
415 Not Useful

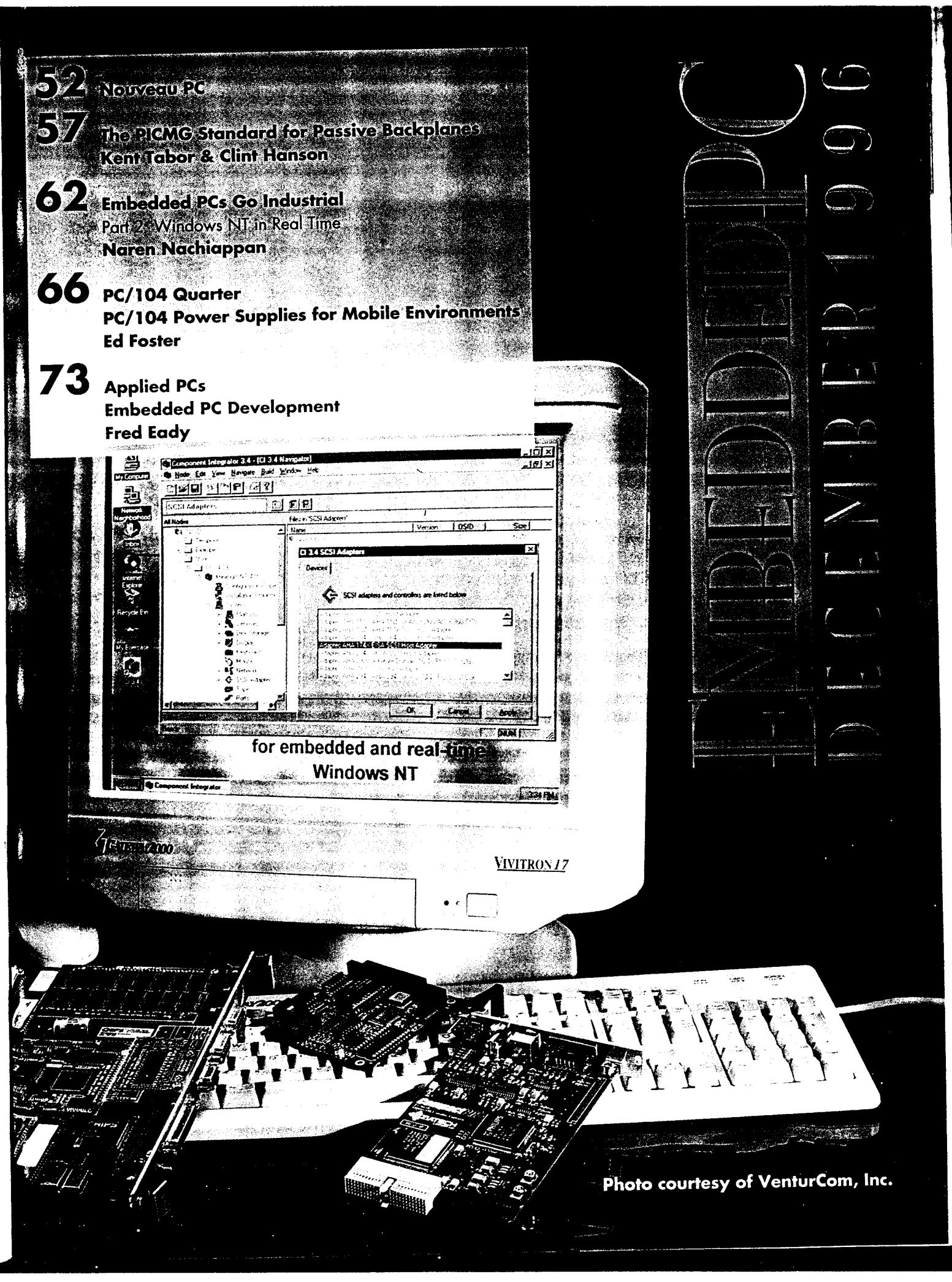
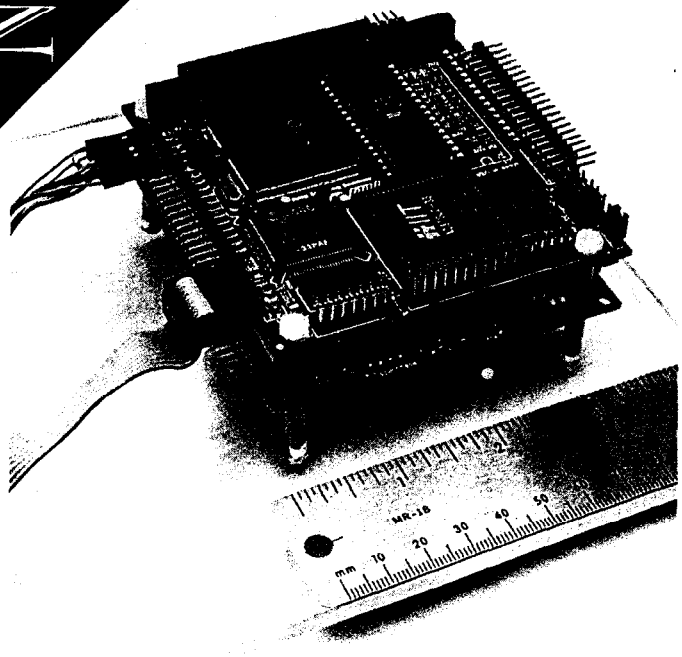


Photo courtesy of VenturCom, Inc.



WEB SERVER FOR EMBEDDED DEVICES

The world's smallest Web server for embedded devices has been announced by Phor tap Software. It is based *on* their Embedded Web Technology, which includes the Realtime ETS Kernel for embedded development, TCP/IP support, and an HTML-on-the-fly package.

The Web server enables developers to make embedded devices universally accessible via the World Wide Web from any PC or workstation with a Web browser. This technology can be used to develop embedded systems for monitoring a wide range of activities via the Internet, including weather stations, seismographic monitors, and floodwatch systems. Alternatively, the technology can be used to develop Intranet applications, such as in-room patient monitors, factory-floor controller monitors, and office systems (e.g., "smart" copiers).

The Phar tap weather station is a working example of the Web server. The demonstration runs on a 4" x 4" '386 single-board computer and provides worldwide access to the local weather outside the company's office via <http://smallest.pharlap.com/>.

The Embedded Web Technology is based on the Realtime ETS Kernel, a 32-bit 'x86-based embedded kernel, which provides a complete development system for building intelligent embedded computing products. The Realtime ETS Kernel is featured in the company's TNT Embedded ToolSuite Realtime Edition development system.

The Embedded Web Technology package is priced at $4995.

Phar lap Software, Inc.
60 Aberdeen Ave. • Cambridge, MA 02 138
(617) 661-1510 • Fax: (617) 876-2972
http://www.pharlap.com/

#510

LOW-COST GRAPHIC LCD KITS

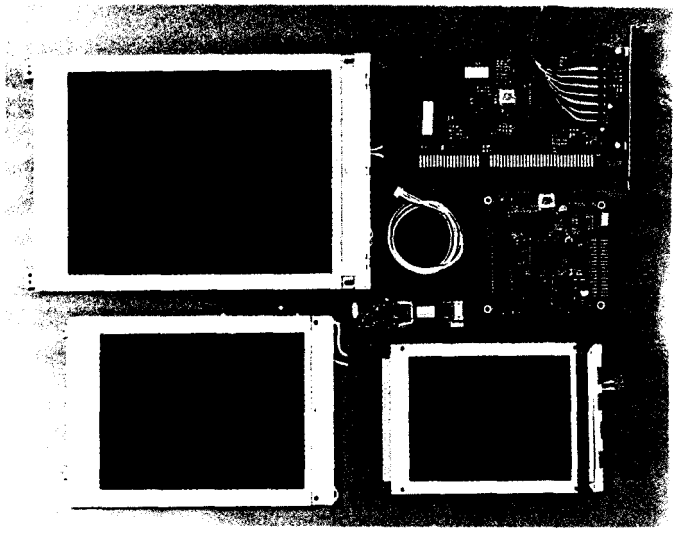
The Simplicity Kit line of passive color and B/W graphic LCDs with PC-compatible ISA- or PC/I 04-bus controllers has been introduced by Apollo Display Technologies. These LCD kits feature high-contrast displays with low-cost prototype and production system implementations for OEM users.

Standard models are available in both color and B/W in several versions-VGA (7.5", 9.4", and 10.4") with 640 x 480 resolution and Q-VGA (5.7") with 320 x 240 resolution. All displays are CCFT backlit and include the CCFT inverters and cable. Designers can choose either an ISA- or PC/I 04-bus controller card with display-matched data cable. All display voltages are generated by the controller.

Sample prices start at $350.

Apollo Display
Technologies, Inc.
194-22 Morris Ave.
Holtsville, NY 11742
(516) 654-I 143
Fax: (5 16) **654-** 1496

#511



Nouveau PC

edited by Harv Weiner

SINGLE-BOARD COMPUTER

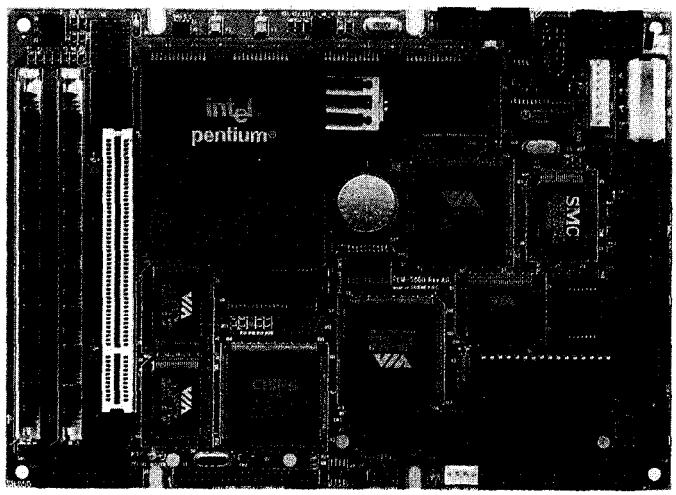
The PCM-5860 is a compact, all-in-one Pentium single-board computer for embedded-PC applications. It accepts Intel Pentium 75–166-MHz CPUs with up to 64 MB of normal or EDO-type DRAM. It features onboard PCI SVGA/flat-panel display, PCI Ethernetinterface (Novell NE2000 compatible), PCI-enhanced IDE controller, and a PCI expansion slot.

Its second-level cache supports both asynchronous SRAM and pipeline-burst SRAM modules with up to 512-KB cache memory. Also onboard are two high-speed serial ports, one multimode parallel port, a floppy-drive controller, keyboard, and PS/2 mouse interfaces. A watchdog timer with time intervals of 1 .6s automatically resets the system. The Award flash BIOS features an advanced power-management system that offers green functions to save power.

The PCM-5860 with a Pentium 75-MHz CPU measures 5.75" x 8" and sells for $680.

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

#512



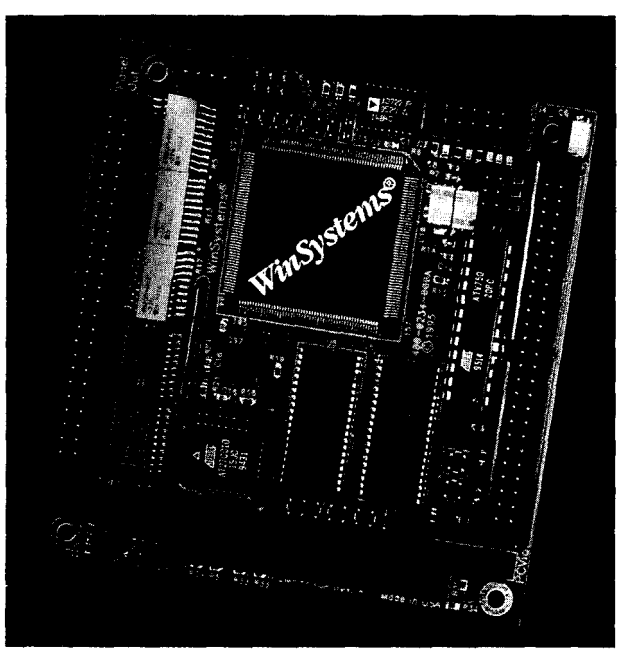
PC/I 04 VIDEO MODULE

A high-resolution SVGA video and flat-panel PC/104 bus controller is available from WinSystems. Ideal for small (3.6" x 3.8") embedded applications, the PCM-FPVGA is an off-the-shelf product that supports multiple LCD panels and CRT displays.

The PCM-FPVGA is based on the Chips and Technologies 65540 high-performance flat-panel and CRT controller. The module supports high-resolution fixed and variable frequency analog monitors in interlaced and noninterlaced operation modes. This module also supports all flat-panel display technologies including plasma, electroluminescent (EL), and active and passive matrix LCDs. It simultaneously supports CRT and LCD operation.

Currently, notwoflat-panel manufacturers' display interface requirements are alike. Different signals, timing needs, and connectors requiredifferent hardware and BIOSs for each flat panel. The PCM-FPVGA solves this problem by defining a special panel-personality module to interface directly with each different panel. The module therefore supports various panels and automatically configures the BIOS for the correct panel type.

Up to 15 PCM-FPVGA boards can be installed on a stack to drive different displays simultaneously, which is useful when a single system needs to display multiple pages of information.

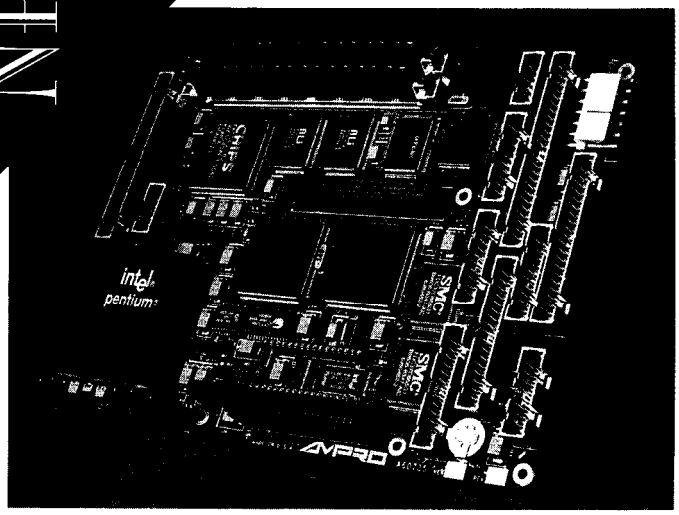


The PCM-FPVGA provides a complete 16-bit PC/I 04 interface. It attaches to any PC/104-bus expansion stack, or it can be added to a larger host embedded single-board computer. The PCM-FPVGA sells for $275 with 5 12 KB of video RAM and **$300** with 1 MB of RAM.

WinSystems, Inc.
715 Stadium Dr.
Arlington, TX 76011
(8 17) 274-7553
Fax: (817) 548-1358
info@winsystems.com

#513

Nouveau PC



PC/**104-*PLUS*** SBC

Based on a 166-MHz Pentium CPU and a PCI-enhanced version of PC/I 04, the little **Board/P5*i*** provides a 25-fold increase in I/O throughput (to 132 MBps) over the standard PC/104 bus. The 5.75" x 8" board is functionally equivalent to a fully populated Pentium workstation. It has four serial ports, one parallel port, floppy and enhanced-IDE drive interfaces, PCI UltraSCSI, 10BaseT Ethernet, and a GUI-accelerated PCI LCD/CRT SVGA display controller. A bootable solid-state disk, ruggedized embedded-PC BIOS, watchdog timer, and other specialized enhancements support the reliability requirements of embedded applications.

Other features include a 256-KB L2 cache, support for Intel's new voltage-reduction technology CPUs, up to 128-MB EDO parity DRAM, APM power conservation, single +5-V supply, and extended temperature operation.

PC/104-*Plus* is a proposed standard that can be used with a wide selection of existing 8- and 16-bit PC/I 04 modules, as well as with newly designed modules that support the standard's high-speed PCI interconnect. A combination of 8- and 16-bit (ISA) PC/104 modules and 32-bit (PCI) PC/I 04-*Plus* modules can coexist in the same stack.

Little Board/P5*i* is priced at $1245 (100 MHz) and $1545 (166 MHz) in OEM quantity.

Ampro Computers, Inc.
990 **Almanor** Ave.
Sunnyvale, CA 94086
(408) 522-2 100
Fax: (408) **720-** 1305

#514

SERIAL PORT MODULE

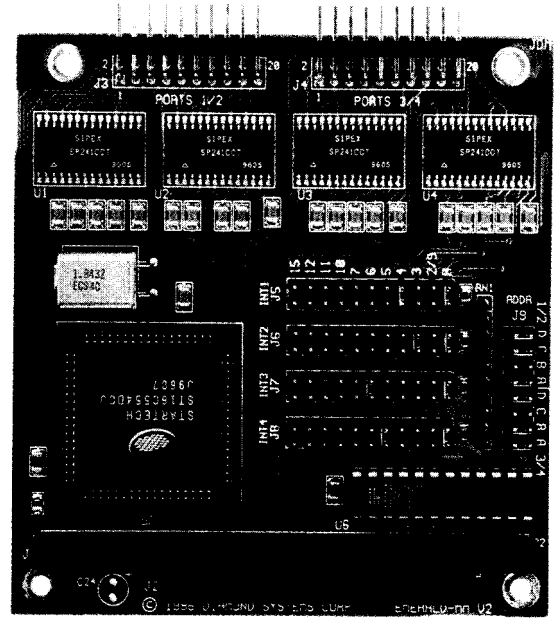
The Emerald-MM PC/104 module for embedded systems provides four RS-232 serial ports operating at up to 1 15.2 kbps. A key feature of the unit is its flexible configuration capabilities. In addition to the standard lower-numbered interrupt levels (2-7) available on the basic 8-bit PC/I 04 bus, the Emerald-MM uses the 16-bit PC/I 04-bus extension to access interrupt levels 1 O-I 5. This reduces the potential for conflict with other system resources and peripheral boards. A total of 10 interrupt levels can be used. The user can also select from among 16 different I/O addresses for the ports, including COM1 through COM4.

The Emerald-MM requires only +5 V for operation and maintains full compliance with all EIA standards for RS-232 communications. The board supports standard operating parameters of even, odd, or no parity, 5-8 data bits, and 0, 1, or 2 stop bits. The four serial ports are made available on two 20-pin headers, with two ports on each header. Multiple boards can be installed in a single system.

The PC/I 04 form-factor board (3.6" x 3.8") sells for $180.

Diamond Systems Corp.
450 San Antonio Rd.
Palo Alto, CA 94306
(415) 813-1100
Fax: (415) 813-I 130

#515



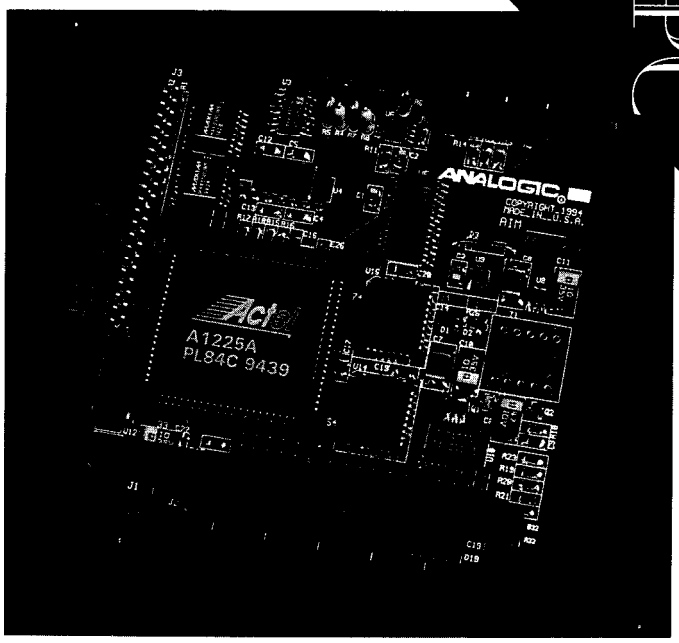
NouveauPC

DATA-ACQUISITION BOARDS

The **AIM16-1/104T** is a 16-bit, 100-kHz analog-input board that provides 85 dB of spurious-free dynamic range (SFDR) with software-selectable inputs of 10, 5, 2.5, and 1.25 V. The **AIM12-1/104T** is a 12-bit, 100-kHz analog-input board providing 75 dB of SFDR with software-selectable inputs of 10 V, 1 V, and 100 mV.

Both products conform to the PC/I 04 format and feature 16 single-ended or 8 differential analog-input channels, 16 lines of digital I/O, flexible triggering options, DMA, and interrupt operation. Both products also feature an extended temperature option that guarantees full specification over a temperature range of -40°C to +85°C. The onboard DC-to-DC converter, powered by a single +5-V supply, provides noise isolation from the system switching.

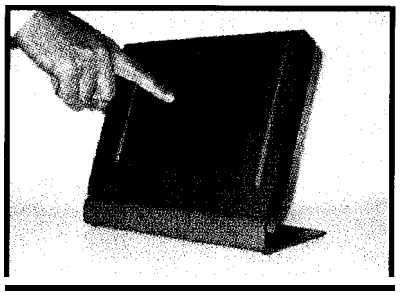
The AIM16-1/104T and AIM12-1/104T are priced at $699 and $499, respectively.



Analogic Corp.
Data Conversion Products Gr.
360 Audubon Rd. • Wakefield, MA 01880
(508) 9774000 • Fax: (617) 245-l 274 **#516**

NouveauPC

Touch The Future



LCD Touch Monitors
LCD Touch Screens
VGA LCD Displays
LCD Controllers
ISA, PC 104, 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

#204

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: **$1** 700 Source Code: add **$1** 000

RTKernel-32

Professional, high-performance real-time multitasking system for 32-bit Embedded Systems.
Supports Intel 386 and higher.
far 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

Kent Tabor &
Clint Hanson

The PICMG Standard for Passive Backplanes

For high-speed bus and CPU performance, you need to satisfy current ISA requirements yet provide PCI support. Kent and Clint show us how the PICMG standard offers this better edge to industrial-PC applications.

Prior to the advent of the PICMG standard, designing full Pentium performance into a passive-backplane industrial PC with support for both ISA and PCI cards was a bit like wrestling a tiger. It could be done, but painfully.

PCs in embedded applications can use either passive backplanes or mainboards. The correct choice is made once operating constraints and specifications are fully defined.

Industrial and Computer Telephony Integration (CTI) applications, for example, often favor passive-backplane systems. In these cases, the passive backplane enables the designer to accommodate the large number of expansion slots necessary to implement such systems.

If the system is highly cost driven without the typical environmental, large interfacing, and performance constraints of embedded systems, a commodity mainboard might be the best fit. ("Commodity" refers to equipment intended for the general desktop-computer market.)

Unlike mainboard systems, however, passive systems are extremely versatile and durable. Contrary to the huge systems needed for CTI, passive systems are also a good fit for small embedded controllers with minimal I/O.

For some embedded applications, a two-card system may be enough to accomplish the objectives without sacrificing the large amount of space needed for the mainboard equivalent. Such modularity enables these passive systems to accommodate backplanes from 2 to 20 slots.

Often, designers looking for such versatility are designing systems for a harsh environment. Hardware designers for passive-backplane systems realize this and design durability into their products. Typically, specifications on products designed for use on passive backplanes are much more stringent than the specifications on mainboard products.

A passive backplane is nothing more than a highway that carries information between the single-board computer (SBC) (see Photo 1) and all of the other devices connected to the backplane. Since the backplane doesn't play an active role in the system's operation, it is referred to as passive. The passive backplane is a PCB populated with connectors only, thereby making them the most reliable component in the entire computer system.

Operational failures rarely involve the backplane. Because of this, repairs to the overall systemcan be taken careofquickly.

In a mainboard system, if the mainboard causes the operational failure, all cards need to be removed from it prior to its replacement. Once replaced, the mainboard needs to be repopulated with the removed cards.

In a passive-backplane system, an SBC failure requires a single-card replacement withoutcompletedisassembly, saving large amounts of service and down time. This also holds true for system upgrades.

Since passive backplanes are intended to be simple and carry many advantages, initial design of an all-encompassing form

proves to be quite challenging. Demanding applications like motion control, vision systems, and video have forced the industrial market to include higher speed buses and CPU performance. With a current data path of 32 bits and a bandwidth of up to 133 MBps, PCI offers the high throughput demanded by the latest high-speed peripherals.

The PCI bus has developed with the Pentium processor in the commodity market to be the dominant bus standard. And, standards that become dominant in the commodity market eventually dominate in the industrial market, too. The lag between the two markets can be attributed to the longer development times associated with industrial components and the consistency that many industrial customers demand.

New development of commodity peripheral boards using ISA and VESA buses is becoming increasingly rare. Although these devices are slowly becoming obsolete in the commodity market, the industrial market needs to accommodate ISA technologies as well as embrace the up-and-coming new technologies.

An example of this transformation to PCI isalreadyapparent in highendvideocards. These inevitable transformations are the primary cause of long-term implementation problems for OEMs and system integrators, who want to minimize change in components yet maintain adequate component availability.

Trying to satisfy current ISA requirements with Pentium performance, while at the same time providing PCI support for the future, makes industrial PCI-backplane design very challenging.

The PICMG (PCI Industrial Computer Manufacturers Group) standard has started to open the door to many new exciting industrial applications of PCI SBCs with passive backplanes. Remarkably, this standard has the support of a large number of industrial-computer manufacturers. This early agreement between such a large portion of the market should result in quicker and more thorough implementation throughout the industry.

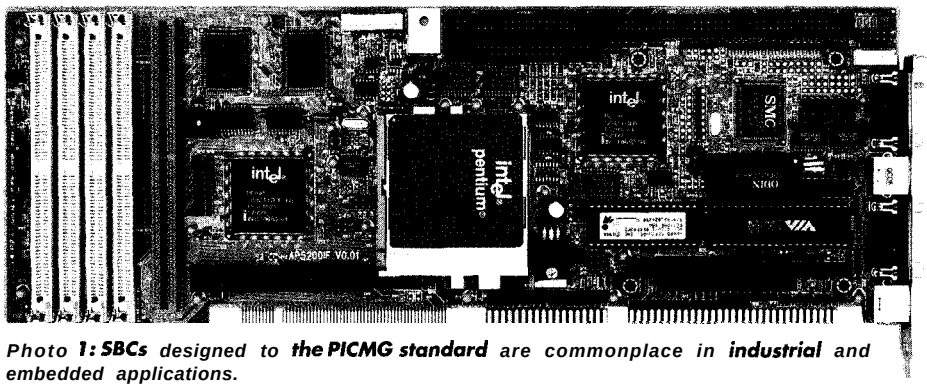


Photo 1: SBCs designed to the PICMG standard are commonplace in industrial and embedded applications.

PCI MAINBOARD LIMITATIONS

As computers get faster and more sophisticated, designers have increased the throughput to peripherals by building on the backboneof theoriginal 8-bit XT (8088) architecture.

The bus architecture evolved from this 8-bit bus to the 16-bit bus of the '286 and '386. This progress was achieved by adding another connector to the 8-bit bus. With the '486 , the 32-bit VESA Local bus (VLB) standard was adopted. The shift to 32 bits was gained by adding a 16-bit connector.

In this evolution, backward compatibility ensured the availability of existing peripheral cards. When the market moved to the Pentium, it faced two options for incorporating PCI performance.

It could replace the existing VLB slots with the PCI bus. This change would force add-on peripheral-board designers to build longer cards. But, that adds cost-a major complaint among VLB peripheral-board manufacturers.

Or, it could place the PCI slots between the existing ISA slots and use the same faceplates. This approach was the one adopted by the marketplace.

Although this idea was great in theory, it added complexity to mainboard designs because it required both ISA and PCI traces to run in the same area. As you see in Figure 1, this required mainboard designers to add layers to the PCB, creating substantial additional cost.

However, most mainboard designers did not stagger the slots. Instead, they put PCI and ISA slots on their boards, thus limiting the number of ISA slots available for most industrial applications.

This arrangement worked well for the commodity-PC market. Customers who specified PCI mainboards usually only required standard I/O peripherals, which are available in ISA and PCI form factors. In the industrial market, this staggered-slot design created new challenges for industrial-PC designers and OEMs alike.

INDUSTRIAL-PC CHALLENGES

Without a technical standard to follow, early PCI passive backplanes required proprietary designs to accomplish the difficult task of interconnecting the SBC to both PCI and ISA peripherals. Since the PCI architecture was designed to be used instead of ISA and to occupy the same space, it was difficult to design an SBC and backplane to accommodate both buses.

A major advantage of the new design was backward compatibility with standard ISA 16-bit backplanes. This capability enables existing applications to upgrade to Pentium performance.

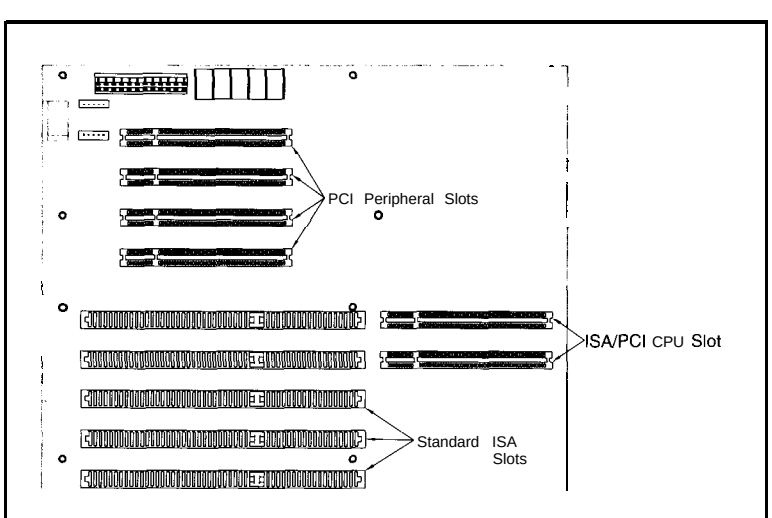


Figure 1: A PICMG-compliant bockplane creates the interface to both ISA and PCI peripherals.

But, without some standard, SBC manufacturers ran the substantial risk of creating a proprietary backplane, only to see a different design adopted as the industry standard at a later date.

CHALLENGES FOR OEMS

Unlike the consumer market where card makers began the move to PCI at the same time as the bus was developed, the industrial market has a 3–4-year replacement cycle. And, without a PCI-backplane standard, there was no compatibility between plug-in SBCs and backplanes from different manufacturers.

Another challenge was balancing the need to use existing, field-proven designs for their applications-rather than adopting riskier (and often more expensive) proprietary technology-with the need to plan for future upgrades in performance.

Also considered was how design longevity should be managed. The product an industrial PC isembedded in may be manufactured in its present form for 4-5 years before its next scheduled upgrade. But, half a decade is an eternity in the PC market, where processing speed doubles every 18 months and new standards are constantly created to keep up with growing CPU and peripheral throughput demands.

Faced with such long end-use product life cycles, OEM designers need to use ISA cards and commodity I/O devices now (most industrial I/O boards don't need PCI performance) to keep costs down and maintain consistency of the purchased product. And yet, they need to provide an upgrade path to PCI backplanes and peripherals.

It's also a challenge to balance the tradeoff between proprietary industrial-PC designs and the commodity-PC market. Because the commodity-PC market is so large, has so much technical development behind it, and is so well-supported, it's impossible for OEMs to ignore it.

The market changes faster than most industrial-product life cycles, and today's cutting-edge standards may be tomorrow's dinosaurs. Clearly, OEMs and industrial-PC designers needed a PCI backplane that could replace proprietary designs already on the market with a single standard.

This standard would allow them to design around current ISA mainboards and I/O cards, yet upgrade to higher performance CPUs (e.g., Pentium and Pentium Pro) at a later date without penalties.

For applications where full Pentium performance is required but ISA slots are needed for compatibility with current I/O boards, it's also desirable to be able to drop a Pentium SBC into an existing ISA backplane. This provides an upgrade path for applications that require the additional computational horsepower of the Pentium PC but don't necessarily need the full benefits of PCI speed and throughput.

THE PICMG SOLUTION

The emerging generation of PCI passive backplanes is based on the PICMG standard. These backplanes bypass these shortcomings and enable integrators of passive-backplane systems to install a PICMG-compatible SBC into an existing ISA backplane (with the PCI connector hanging above the backplane) or into a newer backplane with PCI and ISA slots (see Figure 1). Therefore, OEMs can use popular ISA backplanes and peripherals now and later switch to PCI.

The new PCI passive backplanes better customize the number and mix of ISA and PCI slots needed for each industrial application (see Figure 2). These current-generation backplanes are also better suited to handle the next generation of CPUs, including the Pentium Pro and multiple processors.

For industrial-PC designers, the PICMG standard makes higher performance video and I/O possible, and it has the ability to design around an architecture that's gaining ground as the bus of the future. It also provides opportunities for growth beyond the current maximum of eight card slots.

The PICMG standard is a new standard for card vendors which should take full advantage of the high-throughput PCI architecture. But, aswenoted, industrial-card vendors have been slow to bring noncommodity PCI boards to market.

The PICMG is continuing development of a PCI-to-PCI bridge, which would allow the industrial market to expand the number of possible slots available for I/O cards. But BIOS systems also need to be developed that support PCI-to-PCI bridging.

Current BIOS designs may not see the bridge, and CPU and add-on card bus drivers may not generate enough current to adequately drive the bus. Support for this extension of the PICMG standard is starting, but it probably won't be practical for industrial-PC applications until BIOS manufacturers develop a standard for it.

PCI OR ISA?

So, is a PCI backplane right for your industrial-PC application?

Here are some questions you need to consider when comparing conventional ISA backplanes to PCI backplanes based on the PICMG standard.

Is Pentium performance really needed? PCI is ideal for applications with intense graphics or a large number or rate of calculations, such as high-speed video and

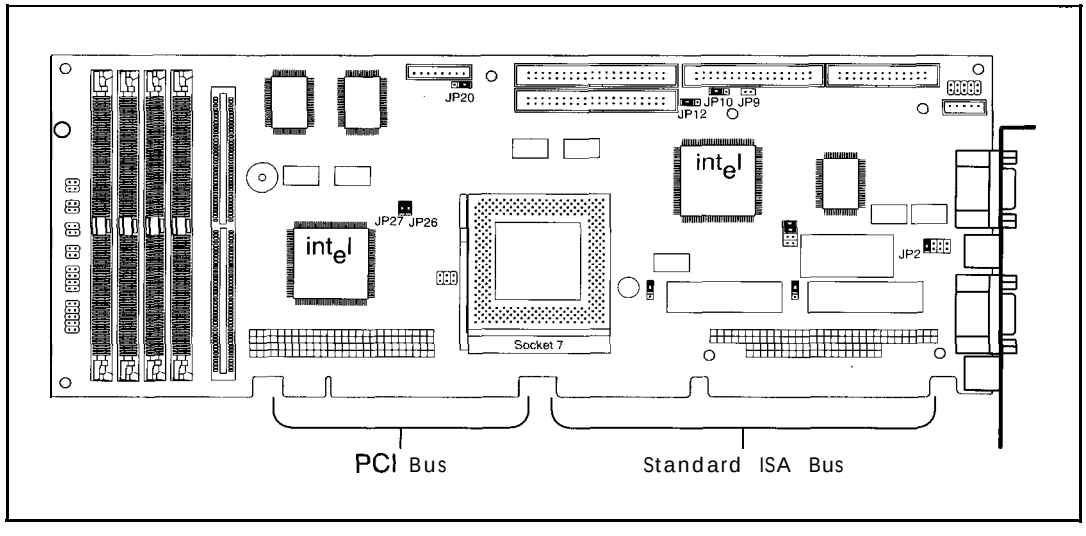


Figure 2: PICMG SBCs can be installed in either ISA- or PICMG-compliant backplanes, adding greatly to their versatility.

high-performance peripherals (e.g., frame grabbers, vision systems, and motion-control/feedback applications).

Is a GUI required? The adoption of better GUIs is driving industrial applications to the PCI architecture. The industrial marketisalmostall Windows based. Many applications are moving to Windows 95 and Windows NT, which require more CPU horsepower and the PCI bus.

Can you get by with a 3–6-month life cycle? If so, a conventional PCI mainboard may be attractive because of its lower cost. In these applications, the industrial-PC designer may view the entire PC as a component and simply exchange it when it fails.

But, this approach may lead to compatibility problems caused by slight differences in mainboard designs and BIOS systems. If you can't get by with such a short life cycle, a passive backplane based on the PICMG standard is the clear choice.

What are your I/O requirements—present and future? If a product with ISA slots has a 3-year life **cycle** but the industrial I/O vendors are planning a switch over to PCI in the next few months, you'll want to seriously consider a passive backplane in the design.

And finally, how many ISA and PCI slots are needed? If a product requires only one or two ISA slots, it may be more cost-effective to stay with a mainboard. On the other hand, if your product uses seven ISA slots now and you're planning to phase them out, a passive backplane based on PICMG provides that flexibility. EPC

Kent Tabor is president of Granite Microsystems. As *an* electrical *engineer with 11 years'* design experience, *he has worked on* development of *SBCs* and industrial *I/O* peripherals, as we// as system-/eve/ design of *SBCs in* machine-control and medical *applications.* You *may reach Kent at tabor@granitem.com.*

Clint Hanson is a *senior* design engineer at Granite Microsystems. He is responsible for new development of *SBCs* and industrial *I/O* peripherals. You may reach Clint at *hanson@granitem.com.*

IRS

416 Very Useful
4 17 Moderately Useful
418 Nat Useful

EPC

Naren Nachiappan

Embedded PCs Go Industrial

Part 2: Windows NT in Real Time

Many *companies using industrial automation need real-time processing, but they also need compatibility with the desktop. Naren suggests that VenturCom's development tools enable* Windows NT to be an effective solution.

In Part 1 (*INK 75*), I described the business and technological trends forcing a major change in industrial-automation control technologies from the older, highly integrated, proprietary architectures to modular, open, embedded-PC architectures.

A critical component of this new PC-based architecture is a robust, open operating system that supports a vast range of commercial, off-the-shelf (COTS) software and hardware modules. Currently, there are only two reasonable candidates for such an OS-UNIX and Windows NT.

While we can debate the relative technical merits of these systems, it's clear that market acceptance, especially in the recent past, is shifting to Windows NT. But, despite its vast amount of COTS software and hardware support and mainstream programmer acceptance, Windows NT isn't a natural fit for industrial-automation applications-especiallysinceitsuffersfrom some critical technical deficiencies.

Crucially, Windows NT lacks real-time support, which I define as the ability of a task to schedule its response to an external event within bounded times. The specific requirements for the bounds on these times are a function of the application.

As I described in Part 1, industrial-automation applications span a range of response requirements-from soft real-time needs of several hundred milliseconds for process control to the hard submillisecond-response needs of motion control.

The standard Windows NT system distributed by Microsoft is capable of, under controlled conditions, reasonably deterministic response rates, which qualifies it as a suitable platform for most process-control applications. However, it doesn't support the next rung of applications, from discrete logic all the way to motion control.

In this article, I'll focus on the enhancements VenturCom has developed to extend the real-time capabilities of Windows NT for use in the entire spectrum of industrial-automation applications. I'll begin with an overview of approaches to implementing real-time functionality **on** nonreal-time OSs. Then, I'll describe the design parameters and give an architectural overview of VenturCorn's RTX approach to enhancing Windows NT with real-time functionality. I'll also discuss the programming interface to real-time functionality, RTAPI, and performance characteristics of the RTX implementation.

After looking at application design in the RTX environment, I'll check out other supporting enhancements that complement RTX. I'll conclude with a discussion of the expected impact of this technology on the automation and related real-time application industries.

REALTIME ON NONREAL-TIME OS

The conceptof making a standard, time-sharing OS-I'll call it ahostOS—real time is not new, either in industry or in academia. RT-Mach (developed at Carnegie Mellon University), VENIX (from VenturCom), Real/IX (from ModComp), and PowerMAX and MAXION/OS (from Concurrent) are examples of real-time OSs derived from existing time-sharing OSs.

A key design criteria for these real-time implementations was to support existing COTS applications designed for the host OS (with varying levels of compatibility), while also providing a real-time environment for embedded applications. The underlying methodologies adopted for such implementations fall into the broad categories of clean-room, layered, dual-OS, and internally reengineered approaches.

In clean-room implementations, a brand new OS is designed from the ground up for real-time performance. This OS shares an API (or API subset) with a host OS, but little else. The OS internals are designed in a manner optimized for real-time performance.

A significant disadvantage of this approach is that compatibility with the host OS at anything other than the most superficial levels is impossible to achieve.

In particular, device-driver compatibility cannot be achieved, necessitating an entire suite of custom-designed device drivers for this type of RTOS.

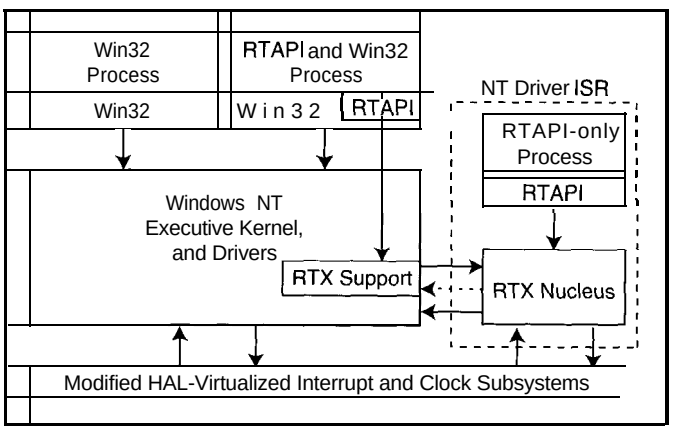


Figure I: This diagram depicts various modules of ***VenturCom's*** hard real-time extensions for Windows NT and the ***intermodule relationships.***

Another disadvantage is the inability to leverage the continuing development of technology in the host OS. Every new technological development has to be reengineered from scratch for the RTOS.

Layered approaches are most typically used by vendors of existing RTOS products (e.g., the UNIX-like API offered by most major RTOS vendors). The basic concept is to provide a library layer that emulates a host OS API on top of the RTOS API.

Needless to say, the layered approach is fraught with peril. Incompatibilities, both subtle and obvious, are a natural result.

However, it is perhaps superior to the clean-room approach because the design radeoff is made in an obvious direction—oward ensuring the best possible real-time performance at the expense of compatibility with COTS APIs.

In a dual-OS approach, two operating systems work together on a single processor. Typically, the RTOS functions as the master OS with control over the hardware assigned to it. Interrupts are vectored to the RTOS, which then processes them and passes them to the host OS.

There are several problems with this approach. The overall stability of the implementation is questionable in most cases, and the performance and stability of the host OS are compromised.

Also, there is no easy mechanism for IPC between tasks running on the two different operating systems. The presence of a dual programming environment with dissimilar APIs forces substantial additional life-cycle costs to be incurred by the developer in terms of initial development and maintenance costs.

In the internally reengineered approach, vendors begin with the source code to the host-OS kernel and modify it to make it fully preemptible and real time. The results (in real-time performance and compatibility) are impressive.

However, the complexity and costs of making itworkand reengineering the imple mentation to match continuing updates in the host operating system is excessive.

Additionally, it isvulnerable to deficiencies in device drivers written for the host OS. Poorly written drivers easily compromise real-time performance, necessitating a careful reengineering of all device drivers in the deployed system.

At one time or other over the past 15 years, VenturCom has developed all of these implementations of RTOSs. Over the years, VenturCom has maintained compatibility with a market-leading OS as a critical design criterion of its RTOS implementations.

Hence, there is a collective knowledge base that has been critical in the design of a new, more enlightened approach to RTOS implementation that combines the advantages of all the above approaches in a novel way to produce a superior implementation.

DESIGN PARAMETERS

We wanted 100% source and binary compatibility to be preserved at all NT interfaces, both documented and undocumented. This requirement, which automatically implies no changes to the NT kernel or to device drivers, ensures that all COTS software packages run unchanged in the real-time NT environment.

As well, the performance had to be comparable to that of the best real-time executive for the same processor. And, minimal reengineering should be needed for compatibility with new versions of NT and the continuing stream of bug fixes and updates from Microsoft.

ARCHITECTURAL OVERVIEW

At a high level, we can characterize the RTX approach in the **following** manner. The approach is based on the principle that application-level partitioning can efficiently split real-time tasks from nonreal-time tasks.

Real-time tasks are typically those engaged in hard real-time control. By definition, they require an API focused primarily on real-time tasks and then rely on normal Win32-based processes to perform all the nonreal-time processing. The major advantage with this approach is that the vast majority of Win32 services need not be modified.

The key additions are a very small set of modifications made to the COTS OS at certain spots-in particular, the points where the OS interacts with hardware-to enable a hard real-time tasking and scheduling mechanism to be implemented.

VenturCom also added a small and powerful set of deterministic IPC mechanisms to allow the RT tasks to communicate with COTS OS tasks.

The basic architecture of the RTX implementation consists of:

- Virtualized Interrupt System (a modified version of the NT HAL [Hardware Abstraction Layer])-allows hardware in-

terrupts to be vectored to special RTX code before being passed on to the NT kernel
- RTX Nucleus-performs scheduling and other support functions
- RTX Support Module-provides support from the Windows NT subsystem for services such as access to shared resources (e.g., memory)
- RTAPI DLLs—implement the API to the real-time system services

Functionally, RTX 4.1 distinguishes between two types of processes-those that only use services provided by RTAPI (designated RTAPI-only processes) and those that use a hybrid of RTAPI and Win32 services or, perhaps, purely Win32 API (designated as Win32 processes).

The fundamental difference between the two is the fact the RTAPI-only processes are run entirely within an interrupt context. They are locked in nonpageable memory and are essentially a part of the kernel context. Context switching to such a thread is, therefore, extremely fast.

The other Win32 processes call a DLL to access RTAPI services. The DLL is implemented at user space and accesses the RTX support module (which is an NT device driver) to implement some of the required functions.

Figure 1 highlights the significant architectural details of the RTX implementation. The start-up phase of an RTAPI-only process is accomplished by a special loader utility.

This loader utility transparently relocates the code into the RTX nucleus environment before passing control to the RTX scheduler. The scheduler then invokes the task.

RTAPI

The RTAPI services function as a self-contained program-supportenvironmentfor hard real-time tasks and as a set of supplemental functions to Win32 processes. In either case, the focus is on the real-time services necessary in a wide variety of real-time applications.

RTAPI defines interfaces to functions for:

- clocks and timers
- fixed-priority scheduling
- I/O bus and physical memory access
- memory allocation and page fault elimination
- device interrupts and interrupt priority setting
- PC with semaphores, messages, and shared memory
- initiation and termination of real-time processes and threads

The syntax of the interface closely parallels existing Win32 system calls, providing a seamless environment in which the appropriate service can be selected by the developer.

Timers	Minimum	Average	Maximum
RTX	4	5	36
DPC	8	27	>1000
Multimedia	24	43	>1000

Table 1: The table shows the reponse latencies for NT timing mechanisms in microseconds. The data was gathered over 30 s on a 150-MHz Pentium system.

PERFORMANCE CHARACTERISTICS

The performance characteristics of the RTX timers were measured against the equivalent performance characteristics of DPC (Deferred Procedure Call) level timers and user-level multimedia timers. A standard PC with a 150-MHz Pentium processor was used as the testbed.

A software mechanism was created to record latencies of each timer by snapshooting the onboard Intel 8254 timer. Minimum, average, and maximum latencies were recorded for each timer mechanism. The data in Table 1 was collected for a single run of 30 s.

The test environment was a standard Windows NT system with a moderate level of disk activity. The RTX timers peak at a maximum of 36 µs for response latencies. The other two timers are essentially unbounded under identical operating conditions.

These performance metrics were derived from an early release of the RTX real-time programming environment. Like any preliminary metrics, they should not be taken as definitive benchmarks. It is quite possible that final performance will be significantly different.

However, the data clearly indicates the success of the RTX implementation strategy in terms of creating a very deterministic real-time programming environmentwithin a Windows NT system.

APPLICATION DESIGN

In the RTX environment, application partitioning enables efficient use of real-time facilities. Real-time applications are typically a combination of the following activities.

Computing typically involvesdata transformation. In closed-loop control applications, feedback from the physical process is transformed to compute new output values to the process.

Control is the physical process or equipment to which the control computer is interfaced and fed new control output. For example, a robot arm is periodically provided with updated trajectory information.

This aspect is typically one which embodies hard real-time requirements. If the robot arm, for example, does not receive trajectory information in a timely manner, it may exhibit a stuttering motion or fail to reach the required destination.

A network interface is a typical requirement of today's control systems. A link to theenterprisecomputersystem allowsonline data to be uploaded for analysis. It allows new process information to be downloaded to the control environment.

Sophisticated operator interfaces are a hallmark of today's control systems. In process-control applications, for example, several hundred process variables may be displayed on compactvideo screens, along with their set points and alarm conditions for easy inspection by factory operators.

Local data logging on storage media is an important requirement of many high-bandwidth data-acquisition applications. The real-time nature of the acquisition makes local storage essential before the reduction and transformation phase.

Thus, the ability to store, retrieve, and otherwise efficiently manage large volumes of data is a critical aspect of such real-time systems.

The design of the RTX implementation of real-time NT requires that users partition the application to separate the real-time versus the nonreal-time processing.

As we see from the analysis of most real-time applications, the tasks they perform naturally map into multiple modules, with the hard real-time operations typically being performed in the control module.

The real-time task responsible for the control activities is required to use RTAPI services, along with RTAPI services. The key to the overall synchronization and

coordination of the application is the deterministic IPC mechanism through which the two categories of tasks communicate.

OTHER ENHANCEMENTS

While these real-time enhancements to Windows NT are critical requirements of many applications, they don't meet the entire range of needs peculiar to industrial-automation applications.

A typical application that needs real-time responsiveness from NT has other needs as well. Two of the more important ones are the embedded operation and development tools.

VenturCom provides Component Integrator 3.4, an embedded application development platform which provides a number of features to support target system development and deployment. CI 3.4 lets you select from over 50 NT service sets (e.g., RAS services, print utilities, etc.) to customdefine a target NT system which is the precise set of services necessary to support embedded applications.

CI 3.4 has a sophisticated knowledge base which incorporates information about interset dependencies enabling the validity of configurations to be analyzed.

Resource requirements of the specified configuration can also be determined by CI. Many options are available for start-up and installation scenarios.

CI 3.4 also incorporates import functions for integrating COTS software into the target-system configuration. Finally, CI 3.4 functions as a repository in which all information about deployed target systems is stored and which can be used to recreate any deployed target at a later point in time.

The following embedded-operation functionality was added by VenturCom as part of its ECK (Embedded Component Kit) suite of enhancements:

- headless operation-operation without a console or keyboard
- flash support-support for popular PCMCIA-based flash media
- minimal footprint support-minimal configuration templates for NT which operate in 8 MB of RAM and support 8 MB of flash

Several other embedded enhancements [e.g., operation from read-only media, diskless operation, and fast boot support) are also under development.

REAL-TIME FUTURE

Such technological development is expected to have a significant impact on the development of the next generation of industrial-automation applications.

This novel approach to the implementation of real-time performance in a mainstream OS with a vast amount of COTS software and hardware support leads to an environment where 100% compatibility is maintained with the host OS.

At the same time, it implements a level of real-time performance that favorably compares with native RTOS implementations.

Traditionally, automation systems have been islands in the enterprise network—islands which required proprietary solutions to communicate with other corporate information-processing facilities. With the implementation of real-time embedded Windows NT, the automation industry can finally break free and benefit from the cost savings integral to COTS-based solutions.

In the short time since the announcement of VenturCom's real-time Windows NT product strategy, leading industrial-automation companies have joined in the development and deployment of this product. In the near future, I expect it will become a standard platform for the development and deployment of industrial-automation applications. EPC

Naren Nachiappan is vice president *for strategic relationships at VenturCom and is directly responsible for sales and marketing initiatives in the Asia-Pacific* region. Naren initiates, develops, and manages critical business relationships *with* major *suppliers, customers, and partners. You may reach him at* naren@vci.com.

SOURCES

Windows NT
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
(206) 882-8080
Fox: (206) 936-7329

Windows NT Development Suite, Embedded Component Kit
VenturCom
2 15 First St.
Cambridge, MA 02 142
(617) 661-1230
Fax: (617) 577-I 605
mds@vci.com

IRS

4 19 Very Useful
420 Moderately Useful
42 1 Not Useful

PC/104 QUARTER

Ed Foster

PC/104 Power Supplies for Mobile Environments

Mobile environments *are* an electronics nightmare. Power surges, *EMI* and *RFI* sparks, loud dumps, *and* environmental factors make it a challenging locale. *Ed,* however, shows us how to stabilize power in the *midst* of the storm.

My car was having a tough time starting, so the mechanic-checked-out the alternator and battery. He yanked the battery cables and connected them to the service computer. Eventually, he replaced the battery, and I left, $95 poorer.

But, after a meeting, I discovered my car fax machine wasn't working. After reciting all of Murphy's laws, I figured out that when the mechanic yanked the battery cables (with the motor running), he interrupted the flow of current into the battery, resulting in a load dump. The electrical energy charging my battery built in magnitude and sought a new way to escape.

My fax machine, which was **on,** couldn't withstand this sudden onslaught, and its power supply was destroyed. Worst of all, I doubt the mechanic will pay for repairs.

A load dump is only one of many sources of power surges in a vehicle. And, these power surges can occur at any time. The vehicle environment is an electronics nightmare, with EMI spraying and RFI sparking everywhere and electrical transients running amuck, zapping the embedded electronics. Electronics in this environment must withstand 600-V transients and load-dump situations (see Table 1).

Although the automotive market is growing about 2% yearly, the amount of electronics introduced in vehicles is much higher. Vehicle electronics are no longer limited to the radio and engine computer. There are now cellular phones, portable computers, fax machines, "smart" navigation with GPS receivers, and car-alarm systems.

Many of these new embedded systems take advantage of PC software. Since PC/104 equipment is smaller, more rugged, and lower in power than its desktop relatives, embedded applications can be quickly and economically brought to market.

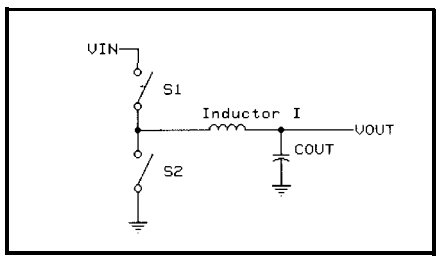


Figure 1: Here's a schematic representation of the "perfect" switching buck regulator.

Designing a PC/I 04 form-factor power supply that runs a PC/104 embedded application and survives the mobile and vehicle environment is challenging. Such power supplies often supply power not only for the PC/I 04 stack but also for external equipment (e.g., flat-panel displays and disk drives).

In addition, the PC/104 size restricts component height to 0.435". All PC/I 04 modules require +5 V, but other modules and attached accessories require +12, -5, and -12 V. A PC/I 04 power supply may only have a total usable PCB area of 9 in? but must supply four different output voltages, operate from 6 to 40 V, supply 50 W, and stay within the small PC/I 04 format.

It's tough, but it's possible with high-performance regulators, MOSFETs, and most importantly, organic semiconductor capacitors.

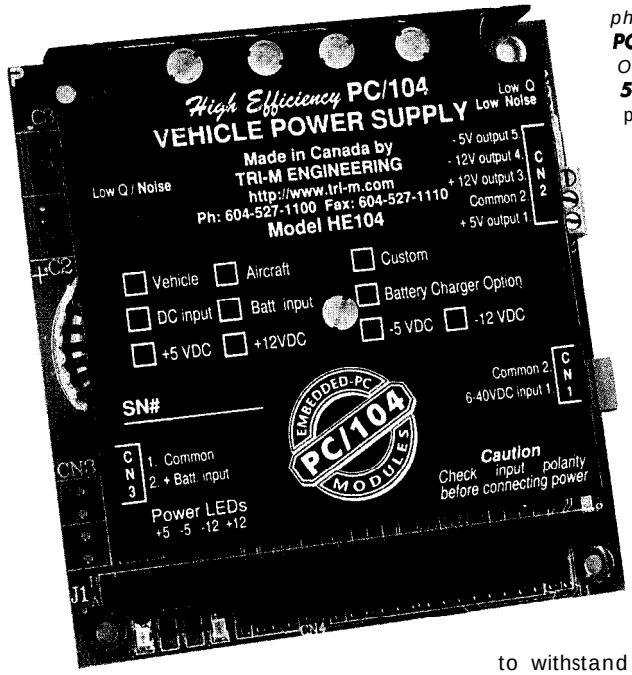


photo 1: ***Tri-M*** Engineering's ***PC/104 vehicle power supply*** user *OS-CON capacitors to* provide a ***50-W,*** high-efficiency ***DC-to-DC*** power *conversion.*

LOAD DUMPS AND TRANSIENTS

toad dumps are an energy surge resulting from disconnecting the battery while it's being charged. The alternator, with a finite response time of 40-400 ms, generates power that has nowhere to go. Thus, an energy surge forms.

The resultant overvoltage is the most formidable transient in the automotive environment. It's exponentially decaying positive voltage, whose amplitude depends on alternator speed and the level of alternator field excitation. It can exceed 100 V.

What makes the load dump dangerous is not just the voltage level, but the amount of energy in it-up to 500 J. Since a joule equals a watt-second, that's a lot of energy to be absorbed in a fraction of a second.

The most efficient and cost-effective method to clamp overvoltage is to shunt the current to ground using a surge suppressor. A surge suppressor relies on the vehicle's wiring and alternator impedance as the current limit. It remains in a high-impedance state until an overvoltage condition occurs. Only use high-energy devices, as 20-50-A peak currents must be shunted.

Several companies (e.g., Motorola, Harris, and Siemens) make suppressors for automotive applications. The Tri-M Engineering PC/I 04 vehicle power supply (see Photo 1) uses Taitron 5KP43A transorbs rated at 5 kW.

Some devices provide zener-diode-style protection, while others have back-to-back zener-diode bidirectional protection. However, unless they're used correctly, they don't protect the electronics.

Ratings on the transient suppressors can be confusing. A suppressor with an avalanche voltage of 24-32 V has a clamp-off voltage of over 40 V.

toad dumps occur infrequently, but mobile electronics must be able to withstand such assaults. toad dumps cooperate slightly though. Their worst-case voltage occurs with higher source impedance, not worst-case source impedance.

In fact, although the total energy of a load dump may be 500 J, a transient suppressor capable of 70 J is usually adequate because of the vehicle's distributed electronics. This holds provided the suppressor avalanche voltage ratings are the same or larger than other suppressors throughout the vehicle.

Quick-thinking engineers take advantage of this and design power supplies and transient protection to withstand higher voltages, which lets everyone's transient suppressors do the work.

ORGANIC CAPACITORS

The organic semiconductor (OS-CON) capacitor developed by Sanyo Electronic Components is an amazing improvement over conventional low electrolytics.

These capacitors provide very lowequivalentseries resistance (ESR), a high ripple-current rating, and they're compact. The OS-CON has about a tenth of the ESR and four times the ripple current rating of electrolytic capacitors.

At low temperatures, the OS-CON advantage is even greater. Since the ESR of OS-CON capacitors is nearly constant from -55°C to 1 05°C, it's ideal for extended temperature operation.

This capacitor is an solid aluminum capacitorwith organic semiconductive electrolyte. It uses the same construction technique as aluminum electrolytic capacitors—a rolled aluminum foil in its element.

The OS-CON differs, though, because in place of the electrolyte solution, an organic semiconductor crystal is impregnated, which uses a highly conductive, solid TCNQ complex salt. As well, it is encased with an impervious epoxy resin, not rubber.

CAPACITOR SIZE SELECTION

OS-CON capacitors are the perfect size for PC/I 04. Their 1 OS-mm height lets a heatsink be installed above them to dissipateany heatproduced bythe MOSFET switches (see Photo 1). The capacitors enable a 50-W, multiple-output DC-to-DC converter with extended temperature operation and in the PC/I 04 form factor.

The filter capacitors are the most important elements in buck-switching regulator design as they supply the instantaneous current requirements and absorb the ripple currents generated. The C_{in} capacitor must

Length of Transient	Cause	Energy Capability (J)	Voltage Amplitude (V)	Frequency of Occurrence
Steady State	Failed voltage regulator	Infinite	+18	Infrequent
3-5 min.	Jump starts with 24-V battery	Infinite	t24	Infrequent
200-400 ms	Load dump from battery disconnection	>10	<125	Infrequent
<320 ms	while at high charging inductive-load switching transient	<1	-300 to +80	Often
200 ms	Alternator field	<1	-100 to -40	Each turnoff
90 ms	Ignition pulse	<0.5	<75	<500 Hz, several times in vehicle life
1 ms	Battery disconnected Mutual coupling	<1	<200	Often
15 µs	Ignition pulse	<0.001	3	<500 Hz continuous
	Accessory noise		<1.5	50 Hz to 10 kHz
	Transceiver feedback		-20 mV	R.F.

Table *1:* Automotive transients can be caused by many factors and have unique characteristics.

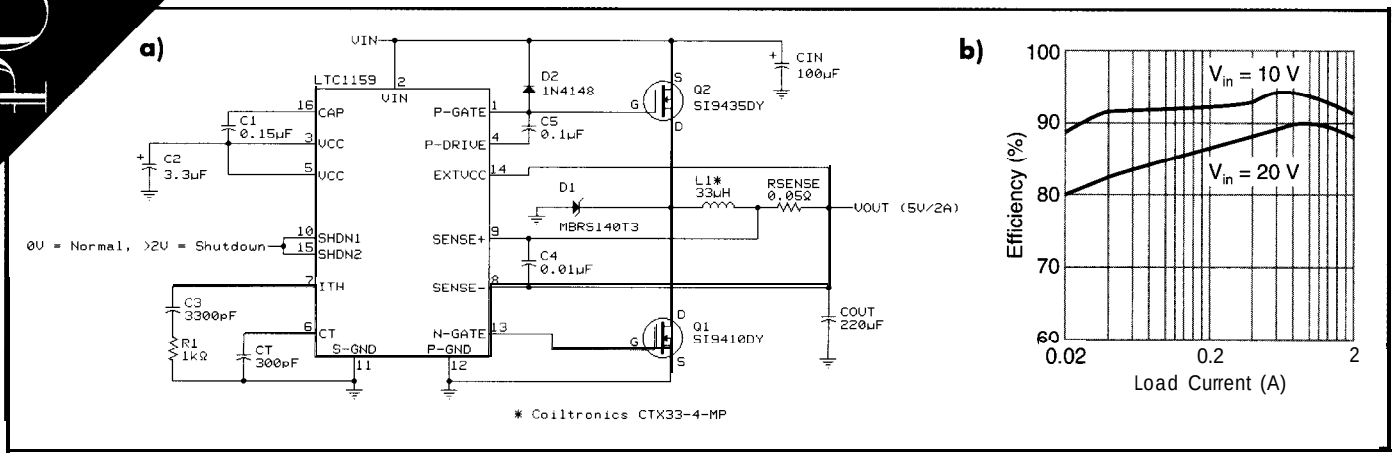


Figure 2: This buck-switching regulator (a) uses linear Technology's ***LTC1*** 159 regulator to achieve a very high efficiency *(b).*

supply the switching currents required for square-wave input currents.

This capacitor must have ripplecurrent ratings large enough to avoid the overheating created by ESR and ripple current. Input ripple current is given by:

$$IRMS = I_{max} \frac{0.5(V_{out}(V_{in} - V_{out}))}{V_{in}}$$

If the input range is 6-40 V, the maximum ripple current occurs when:

$$V_{in} = 2V_{out} \text{ or Max } IRMS = \frac{I_{max}}{2}$$

Therefore, if the supply has a maximum output of 10 A, an input capacitor with a ripple-current rating of 5 A is required. Capacitors can be paralleled to achieve the required ripple-current rating.

The selection of C_{out} is driven by the need for output-ripple voltage. The output ripple is determined by both the inductor value, output capacitor, and frequency.

You can calculate the ESR for any output-ripple level knowing the inductor and the frequency set by the regulator IC. Notably, in C_{out}, the output-ripple voltage is independent of output current.

For a wide input-range switching regulator, the worst-case ripple voltage can be approximated by:

$$V_{p-p} = \frac{ESR \times V_{out}}{L \times f}$$

Transition Loss (W)	V_{in} (V)	Efficiency Loss (%)
0.13	16	0.5
0.39	28	1.5
0.80	40	3.2

Table 2: *Here's a* comparison of transition losses to input voltage with fixed-output operation (I_{out} = 5 A, C_{rss} = 200 *pF,* freq *[f]* = 100 *kHz, and* V_{out} = *5 V).*

where L is the inductor and f equals the frequency of the regulator. If V_{p-p} needs to be less than 25 mV, L is 50 µH, and **f** is 100 kHz, then ESR equals 0.025 Ω.

The microfarad capacity of the capacitors is not used in the formulas. At frequencies of 10 kHz and above, a capacitor's total impedance is almost identical to ESR. Its ESR is the design element used in switching-regulator equations.

EFFICIENCY AND SWITCHES

The success and popularity of PC/I 04 is largely due to readily available software and standard development tools for the PC architecture. PC/I 04, however, requires much less power than its desktop cousin. The embedded PC/I 04 application may interface with other accessories (e.g., a flat-panel display) or additional power requirements.

Operating a linear regulator is impractical due to the large heat dissipation required, but a buck-switching regulator operates from 6 to 40 V. It's 90% efficient over its entire range. Thus, a PC/104 embedded system with a load of 40 W needs to dissipate 4 W of heat.

Figure 1 shows the "perfect" basic buck regulator consisting of two switches, an inductor, and an outputcapacitor. Switches S1 and S2 open and close alternately so the voltage applied to inductor L can be either V_{in} or 0.

The DC output voltage is then the average voltage applied to I. V_{out} is V_{in} multiplied by DC, where DC is the duty cycle.

The perfect switching regulator doesn't dissipate power in converting one voltage to another. It's 100% efficient. But, inductors, switches, capacitors, and circuit board wiring aren't perfect. I^2R power is lost.

Total power loss = $L_1 + L_2 + L_3 + \ldots$

where I_1, I_2, I_3, and so on are the individual losses from each component.

The choice of switches for the buck regulator greatly affects its efficiency. Figure 2a shows a typical buck synchronous switching regulator with MOSFET switches. Figure 2b shows the high efficiency achieved by this register.

To achieve an efficiency of 90%, the MOSFET switches must have a minimum power loss. Choosing a MOSFET with a very low R_{ds} results in large transition losses, but a MOSFET with a high R_{ds} results in large I^2R losses.

Transition losses (see Table 2) can be estimated by:

$$\text{Transition loss} = 5V_{in}^2 \times I_{max} \times C_{rss} \times f$$

I^2R losses can be easily predicted because, in continuous mode, the output current is chopped between the two MOSFETs. If the two MOSFETs have the same R_{ds} then the resistance of one can be summed with the resistances of the inductor, PCB traces, and current sense resistor.

If MOSFET R_{ds} equals 0.028 Ω, then inductor DC resistance is 0.02 Ω, sense resistor equals 0.02 Ω, and PCB traces area 0.01 Ω. The total resistance equals 0.078 Ω (efficiencies calculated with V_{out} = 5 V) as shown in Table 3.

Amp Output	I^2R Losses (W)	Efficiency Loss (%)
1	0.078	1.56
2.5	0.487	3.89
5	1.95	7.8

Table 3: PR losses are a function of load current.

Table 4: It's important to keep in mind **the** individual losses **for a** high-efficiency 25-W **buck-switch-**ing regulator.

I^2R Losses	Transition Losses	Core and Capacitor ESR Losses	Total Losses
7.8%	0.5%	~2%	10.3%

Choose a MOSFET-package style that dissipates the generated heat. A SMD-220 surface-mount package has a thermal resistance of 40°C/W mounted on a 1 in? PCB.

Using an SMD-220 at an elevated ambienttemperaturelimitstheamountofpower that can be handled. Smaller parallel MOSFETs help dissipate heat, but with the PC/I 04 form factor, this isn't practical. A TO-220 package is better because a heatsink provides better thermal resistance.

Other losses, including C_{in} and C_{out} ESR dissipative losses, Schottky losses during dead time, and inductor-core losses, generally account for 2-3% total additional loss. Total losses for a 25-W **supply** operating at 16-V input are shown in Table 4.

WIDE TEMPERATURE OPERATION

Unlike your air-conditioned office, mobile and vehicle environments have temperature extremes of -40 to 50°C. Filter capacitors and MOSFETs are most affected by temperature extremes, which reduce component life and decrease performance.

Traditionally, low-ESR electrolytics were used with switching power supplies. At normal ambient temperatures, their performance may be adequate. But at –40°C, their ESR may increase by well over 10 times as shown in figure 3.

In a buck converter, output-ripple voltage is proportional to C_{out} ESR. A drop in ambient temperature may result in so much ripple voltage that the PC/104 modules shut down or act erratically.

At high temperatures, electrolytic capacitors have a very limited life expectancy. At 105°C, they are rated at 2,000 h. Life expectancy doubles for every 10°C below 105°C, so that at 65°C operation, the expected life is 32,000 h. At 85°C, the expected life is reduced to 8000 h.

The life expectancy of OS-CON capacitors is typically rated at 2000 h at 105°C (the long-life SH series is rated at 5000 h). However, the life expectancy increases 10 times for every 20°C drop. Thus, they have an expected life of 200,000 h at 65°C and 20,000 h at 85°C.

High temperature also negatively affects MOSFET switches. The MOSFET R_{ds} typically rises by 50% with an increase of 80°C in junction temperature. At elevated ambient temperatures, it's difficult to dissipate heat and the resulting higher R_{ds} causes them to generate even more heat.

This additional heat has a direct impact on the efficiency of the power supply. Fortunately, MOSFETs don't mind the cold.

Even with 90% efficiency, heat must dissipate. If the power-supply module is in a PC/104 stack, forced ventilation is required.

By placing the power supply on the stack's top, a heatsink with approximately 10 in? can be used for a thermal coefficient of about 10 C/W. Also, if the heatsink is in contact with an enclosure wall, heat is dissipated for the enclosed environment.

CAPACITORS-CHEAP INSURANCE

In designing PC/104 power supplies for the mobile and vehicle environment, reduce the length of traces carrying heavy currents (see Figure 4) and use separate power and signal grounds. The internal reference voltage of the regulator IC is referenced to ground. Any error in ground-pin voltage is multiplied at output. To ensure good load regulation, thesignalground

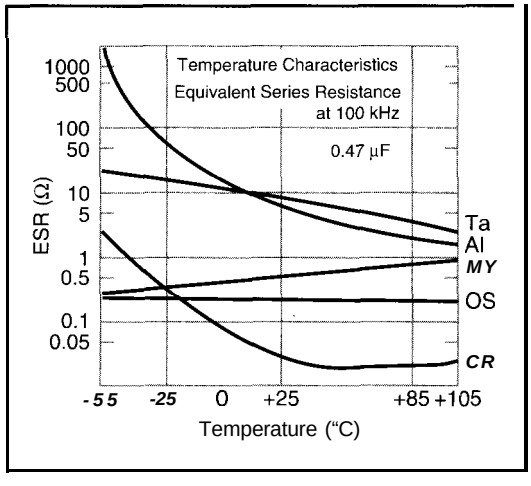


Figure 3: Depending on how a capacitor is made, ***it's more suitable for particular applications. As you can see, each capacitor type has different ESRs over a range of temperatures (Tantulam = Ta, Aluminum = Al, Film = MY,*** and Ceramic **= CR).**

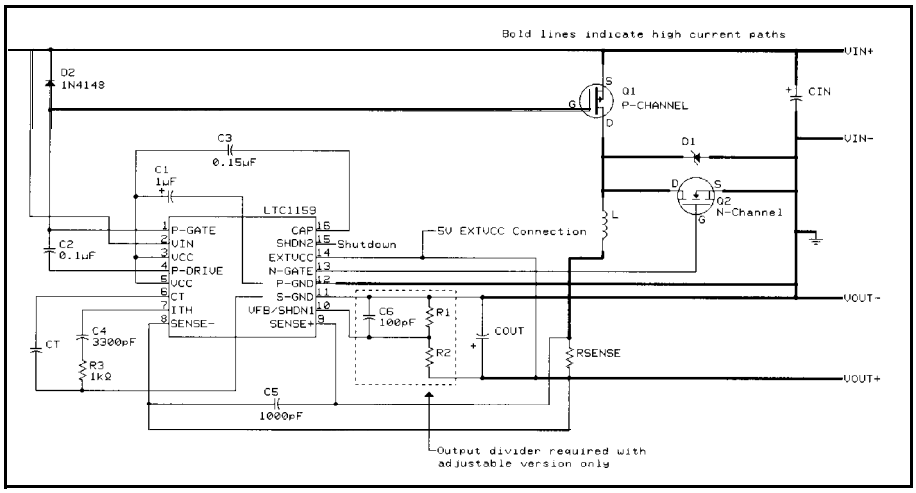


Figure 4: Note the high-current paths in this buck-switching regulator.

pin of the regulator IC directly connects to the output node or capacitor.

The best way to deal with heat dissipation is to not produce heat. While 100% efficiency is not attainable, better than 90% is. With high efficiency, only a moderate-sized heatsink of 10 in? is necessary.

If the application draws 15 W or less, a heatsink may not be needed. PC/104 module stacks aren't designed for optimum heat dissipation. However, by placing the power supply at the top of the stack, heat dissipation is reasonably good.

The OS-CON capacitors are revolutionizing the filtering of switching regulators. They have a very flat-temperature ESR graph from -55°C to 105°C, very low ESR, and a high ripple-current rating. **For** example, a 6SH330M (0.4" x 0.4" radial OS-CON, 6.3 V, 330 µF) capacitor has an ESR of 25 mΩ and ripple rating of 2.5 A at 85°C.

OS-CON capacitors run $1.25-1.50 each in 1000-piece quantities. But with their low ESR and flat temperature curve, fewer are needed. On a dollar-per-ESR basis, the OS-CON is an even better buy.

Compact OS-CON capacitors reduce the overall size of the switching regulator. At full switching current, the switching transients at the input MOSFET get large.

Tests indicate the inductance from board traces can cause inductive spikes of about 2 V per inch. These switching spikes can cause erratic operation.

When I'm asked about the three most important elements in a good switching regulator, I answer, "Capacitors, capacitors, and capacitors."

There's more cost in the capacitors than in any other component of the design, but they have the greatest impact on power-supply performance, and they determine its life expectancy.

Any vehicular power supply is a hazard to itself and the equipment it powers—unless attention is paid to high-energy transient suppression. Protection costs $2-5, but it's cheap insurance! PCQ.EPC

Ed Foster has over 20 years' *electronic design experience. He cofounded Tri-M* Engineering, a leading supplier of power supplies *for the mobile and vehicle* embedded-control market. You may reach Ed at *efoster@tri-m.com.*

SOURCES

OS-CON capacitors, **6SH330M**
Sanyo Video Components
2001 Sanyo Ave.
San Diego, CA 92 173
(619) 661.6835
Fox: (619) 661-1055

United Chemicon
625 Columbia St.
Brea, CA 9282 1
(714) 255-9500
Fax: (714) 255-9400

Sprague Electric
P.O. Box 23 1
Sanford, ME 04073
(207) 324-4 140
Fax: (207) 324.7223

Linear Technology
1630 McCarthy Blvd.
Milpitas, CA 95035
(408) 432-I 900
Fax: (408) 434.0507

Tri-M Engineering
1301 Ketch Ct., Ste. 6
Coquitlom, BC
Canada V3K 6X7
(604) 527-I 100
Fax: (604) 527-I 1 10

IRS

422 Very Useful
423 Moderately Useful
424 Not Useful

Applied PCs

Fred Eady

Embetlded PC Development

Using Visual C++ and the Intel '386EX evaluation board, Fred has a blast showing us how Phar lap's TNT Embedded Too/Suite is assembled and how applications interface with it. It represents the birth of a new and powerful tool.

You—yes, you!-were born to be wild for embedded PCs.

WHAP!! Gasp! You are breathing... you've just been born ("spawned" if you're a UNIX type or a fish). For the past nine months, your embedded biowore has prepared you for this moment.

This is it.. .lifeus interruptus, power-on-reset time. Welcome to the new world, homey.

All that time **you** were in the womb, your embedded biosystem was operating in what we embedded heads call "real mode," building and initializing all your bodily functions. Those of us that follow the way of the embedded know this world as "protected mode." Your PE bit was set as soon as you gasped for that first breath of air.

Close by, parental biosystems (i.e., hosts) use their internal counters and I/O ports to confirm that **you** have the correct number of toesandfingers. Yourfirstindirectchecksum calculation!

And, as if life isn't tough enough already, the nurse is calling you "cutie pie."

An external interrupt (the doctor's gentle and loving slap on your butt) causes an embedded program within you to kick off. You scream, announcing to your hosts that your processor clock is running. All you know is that you require input-now!

Youremb edded biowarereal-timeclock has initialized and started to run, and interrupts are coming in at a fantastic rate from no less than five input sources you didn't even know you had a few minutes ago! Your highest priority is to establish contact with a host and download sustenance for your brand-new embedded biosystem.

Soon enough, you link up with your host, service the pending interrupts, and produce your first output. It's a little messy, but at least the ports are operational.

Over time, your embedded bioware becomes more capable and begins to handle interrupts and I/O with ease. Input and output become logical functions. You can handle repetitive tasks once taught to do so. In other words, you get programmed just like your firmware-laden embedded-PC counterparts.

You were bred to compute, analyze, and react. Your embedded intelligence can now run multiple program sequences simultaneously.

In time, you find you can control complex machinery. You're even capable of running programs dependent on your surroundings. Just like **your** firmware first cousins, you'll do these things and much more up until, at last, your power is removed.

RIGHT TOOL FOR THE JOB

How did all of your biological perfection come about? Simple. Your Maker has a very fine bioware development toolkit. Good tools produce good products consistently.

APC

Dr. Frankenstein is the only other gentleman I know that also possesses such tools. But, Dr. Frankenstein was not available for consultation.

Obviously, you and I cannot purchase a bioware toolkit. But, from an embedded-PC standpoint, with the right set of tools, we embedded programmers can come real close to being divine.

LIFE IN AN EMBEDDED WORLD

Building embedded-PC applications is a logical process involving a fairamountof trial and error, pain and suffering. The pain involved with debugging and final test is relieved somewhat by a good set of embedded-PC firmware tools. In most situations, the hardware is a fairly stable '386/'486/Pentium-based system that presents a minimum of hassle.

Judging by the previous paragraph, what do you think we're doing today? You guessed it. We're constructing embedded firmware using Phar Lap Software's very fine embedded-PC development toolkit.

TNT EMBEDDED TOOLSUITE

Phor Lap's Embedded ToolSuite (ETS) is primarily aimed at the C programmer, but it's also useful to the hard-core 80x86 assembler programmer. ETS comes in two flavors-standard and real-time.

The ETS Realtime Edition adds features to the Standard Edition that involve developing real-time multitasking 80x86 embedded systems. For now, I'll concentrate on using and understanding the Standard Edition. The ETS replaceable-module architecture is the same for both editions.

The TNT ETS Standard Edition enables the embedded-PC programmer to build 32-bit applications using any 32-bit C or C++ compiler offered by Microsoft, Watcom, or Borland. These '386/'486/Pentium-based applications run under control of the ETS kernel. As a bonus, CodeView, 386ISRCBug, or Turbo Debugger can debug your embedded application.

The added value the TNT ETS brings to your keyboard is that you, the embedded programmer, can assemble and debug embedded programs much like you do for desktop-PC applications. With the exception of ETS, you can use the same tools for embedded and desktop development.

To successfully run the ETS environment, you need:

Listing I: *It* ain't complicated, but it works.

```
#include <studio.h>
#include <stdlib.h>

void main0
{
  int i;

  printf("\n   ***CIRCUIT CELLAR APPLIED PCS ***\n");
  printf("\n   ***Phar Lap Embedded ToolSuite ***\n");
  printf("\nFLASH code entered through BIOS extension method.\n\n");

  for (i - 0; i < 10; i++){
    printf("It doesn't have to be complicated to be embedded...\n");
  }
  printf("\nThis is what it's all about!\n\n");
  printf("\nWE'RE OUTTA HERE...\n");
  exit(0);
}
```

- '386 or higher host PC
- host serial ports COM1 and/or COM2
- host parallel ports
- LPT1, LPT2, or LPT3
- 4 MB of host memory
- 20 MB of host free disk space
- MS-DOS V.3.0 or above (host resident)
- 32-bit C or C++ Compiler by Microsoft, Borland, or Watcom

That's all the TNT ETS requires.

Since the ETS environment targets the industry-standard PC/AT architecture, you can even tie two desktop mongrels together with a null modem cable and use the TNT ETS to develop 80x86 embedded applications. There's even a 32-bit TNT DOS-Extender that lets you begin embedded development before the target hardware walks through the door!

Although the PC-to-PC hookup has its strong points, most hired embedded guns shoot from the host PC at an embedded version of the PC/AT standard sitting at the opposite end of a communications cable.

Speaking of cables, the TNT ETS comes equipped with industry-standard Laptink parallel and serial cables. These are great! The genders of the serial cable are correct (for a change), and both 9- and 25-pin connectors are placed at each end.

I don't know about you, but I hate wiring null modem cables and then having to pull out the data scope to shoot the wiring bugs.

ETS KERNEL

for a programmer, there's nothing worse than a totally dumb embedded system. Poorthings, youcanonlyfeel sorryforthem when you take them out of the antistatic bag and find that they don't have a boot ROM or BIOS.

Usually, you take them in and take the time to code a simple monitor just to get them to the point where they can hold up their heads and speak. If you don't have precanned monitor code in the cupboard, you spend precious time writing and debugging a custom monitor before beginning your real work. Depending on the complexity of your embedded peripheral subsystem, this could take a while.

The ETS kernel eliminates the need to spoon-feed a brain-dead embedded micro and its associated peripherals. It initializes the target **embedded** system to run in 32-bit protected mode and provides a foundation to support a C/C++ run-time library.

The ETS kernel automatically sets code and data selectors to a flat memory model with 4 GB of address space. In addition, the kernel can be configured to provide host communications support, which establishes a link between the host and target systems via the serial or parallel ports.

These features provide any programmer familiar with MS-DOS and Windows development tools a means of easily developing 80x86 embedded firmware. Since the ETS system is based on replaceable software modules, it adapts to almost any 80x86 embedded system.

LECTURE OR LAB?

Once again, it's decision time. The Phar Lap TNT Embedded ToolSuite documentation stacks in at approximately 1' including the companion Intel tech manuals.

In the past, I've asked for a show of hands about how to present all this knowledge. But, since we've all just suffered through yet another big election year, I'm giving our president the opportunity to make this major decision for you. After all, he is an elected official sworn to serve the people.

I spoke to him briefly-he's a very busy guy-and he stated that he'd rather see a hands-on approach than an academic treatise when it comes to embedded-systems programming tools. Thank you, Mr. President.

EMBEDDED BUILDING BLOCKS

Don't get me wrong. Phar tap's ETS documentation is excellent. It iswell-written and caters to the beginner as well as the seasoned embedded engineer.

But, the best way to understand the ETS architecture is to break it down into individual modules. Once you understand how one particular module interrelates to others, you feel the power the TNT ETS exudes.

Modularity and replaceable software modules are the pillars that enable ETS to wield this power. The TNT ETS is composed of five major components-the ETS kernel, Visual System Builder, Linktoc (32-bitlinker/locator), shells for embedded cross-debugging, and C and C++ run-time support.

We already have a pretty good idea about what the ETS kernel does, so let's get with it, build our own ETS kernel, and run an application under it.

OUR EMBEDDED ENVIRONMENT

Man has been to the moon, women have set records in space, the B-2 flying Wing is reality, probes are reaching the atmospheres of remote planets, and aluminum cans have pop tops.... So, why write a complicated program to demonstrate our ETS kernel? I won't.

The object here is to show you how the ETS kernel is assembled and how applications-both simple and complex-interface with it. I'll be working with Microsoft's Visual C++ Compiler V.4.0, targeting an Intel '386EX hardware platform. The test code is shown in Listing 1.

VISUAL SYSTEM BUILDER

Visual System Builder is a Windows-based application that ships with the TNT ETS. VSB is essentially an interpreter that interfaces you, the programmer, with the linker/locator Linktoc.

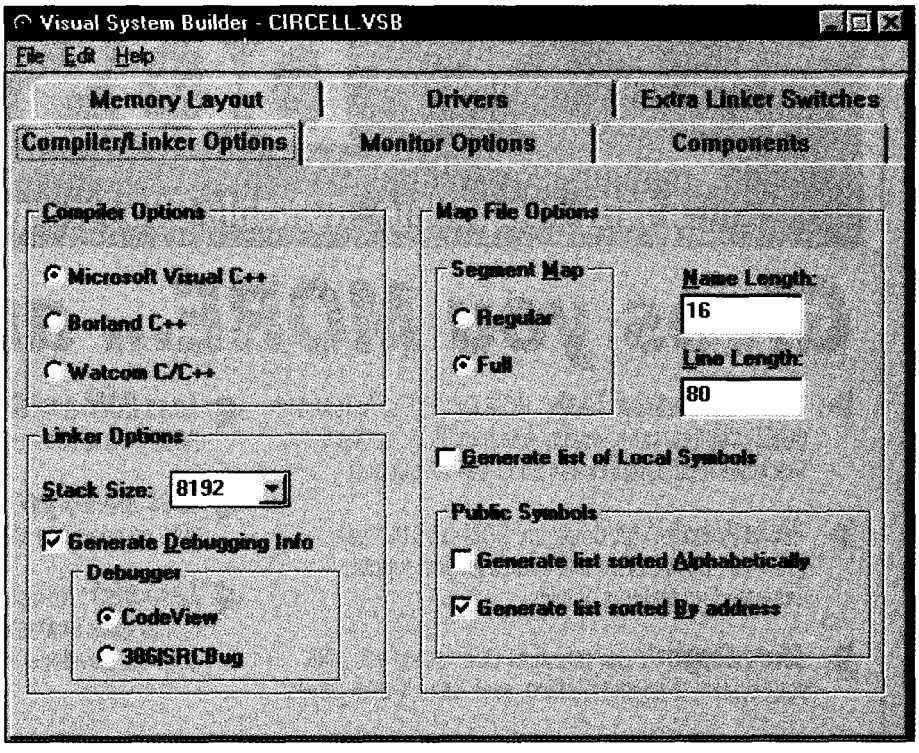


Photo 1: The first step toward a workina ETS Monitor is a default debugging setup for the Microsoft Visual C++ Compiler.

The idea is to eliminate memorization or misinterpretation of the numerouscommand line switches needed to build a custom embedded application with LinkLoc. When you make the VSB choices for your hardware target, VSB writes this information to linker command files which Linktoc references when you build your application.

VSB allows the system builder to choose one of three predefined hardware templates. These templates determine what replaceable-module configuration VSB loads.

The selectable hardware platforms includea generic PC/ATcompatible, a Forth-Systeme Module '386EX board, and an Intel '386EX evaluation board. I've selected the Intel board as the target.

Photo 1 shows the VSB property sheets. Let's eliminate the default sheets and study the rest. The Compiler/tinker defaults to the Visual C++ and assumes we want debug values set accordingly.

The Components sheet relates to the Realtime Edition. Nothing to fill in there.

The Drivers tab takes us to PC/AT-compatible land, and that's where we want to be. The Extra tinker Switches tab allows input from VSB to each individual linker file.

These areas are useful for overriding switches in the linker command files. Since I won't be taking advantage of that feature either, that leaves Monitor Options and Memory Layout for this application.

One of the parameters presented when selecting the Monitor Options tab is Boot Method. The three options include Boot Diskette, Boot Jump ROM, and BIOS Extension ROM.

The '386EX evaluation board can't host a diskette drive, so scratch the Boot Diskette option under Boot Method. The Boot Jump ROM method assumes that no BIOS is present to initialize the target chip set.

Grammar Engine's Forth-Systeme board is an example of an embedded platform that can employ the Boot Jump method since it comes with no installed boot devices. I could also wipe out the BIOS on the Intel board and use the Boot Jump method there, but in this case, there's no good reason to.

Hosing a perfectly good BIOS is like bailingoutofa perfectlygoodairplane.The logical choice is to stay in the plane and choose the BIOS Extension ROM method.

By making this choice, I'm using BIOS capabilities that let me be found after the BIOS has completed its power-on, self-test, and initialization routines. BIOS searches for a signatureofthree bytes-Ox%, 0xAA,

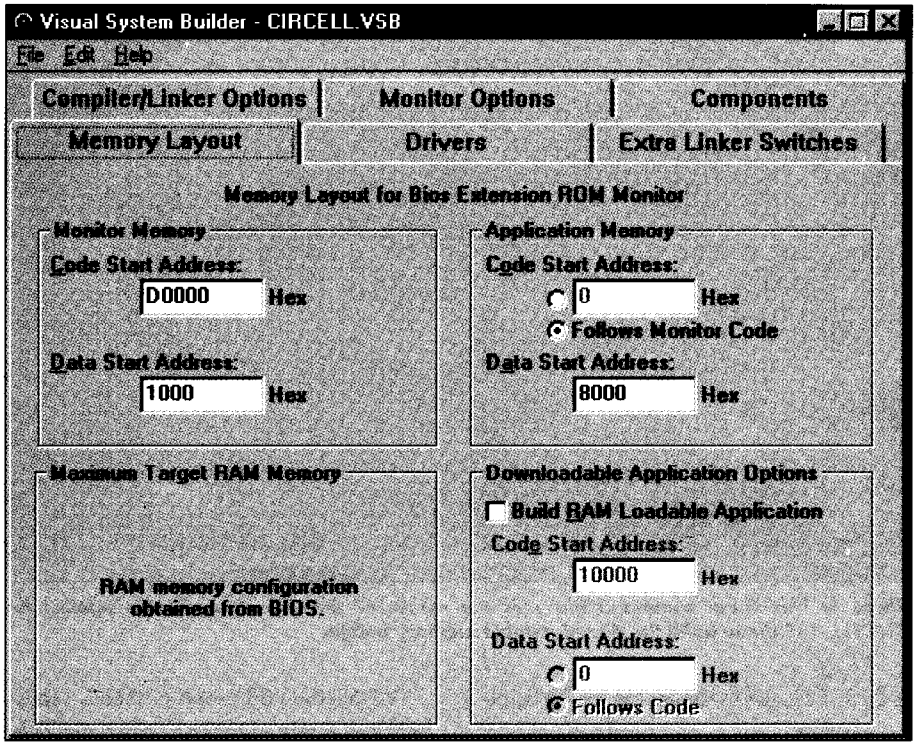


Photo 2 As you can see, you have full control of the memory areas. Note the *option to* build a downloadable RAM-resident *image as well.*

and the number of 512-byte blocks in ROM.

Normally, after the BIOS search, the BIOS looks for the "Ox55 0xAA" pattern starting at location 0xC0000 and continues searching in 2-KB increments through address 0xDFFFF. An addition look is taken at 0xE0000. The search at OxEOOOO assumes the entire 64-KB area will be used.

In our case, the ETS Monitor is booted instead. Our kernel is in ROM, and control is passed to the beginning of the code.

The Interrupt Stacks area is a VSB default and should work well with our small application. Note that the Monitor interrupts are disabled, allowing cleaner and faster downloads from the host during debugging. I could enable interrupts, but the code downloads are painfully slow when the embedded system is servicing application-oriented interrupts.

Under the loader/Debugger, I want to select options that let the kernel communicate via COM 2 at the highest possible

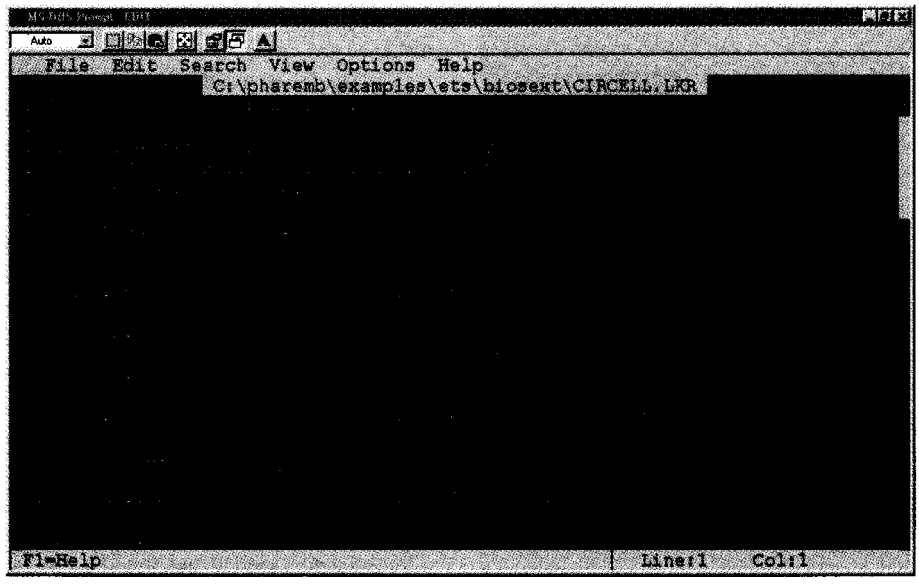


Photo 3: Everything *I* told ***VSB*** about the ETS Monitor configuration is here.

speed with the host. COM2 is the load port on the Intel system. The "Wait at startup for host" box tells the kernel to wait for a signal from the host before starting the application. Otherwise, the kernel starts the application immediately after gaining control from the BIOS. The start signal is issued by the cross-debugging shells (CVEMB and RUNEMB for the Microsoft Compiler).

The Memory Layout property sheet is shown in Photo 2. I don't want to play hide-and-seek with the BIOS, so take note that I have placed the code start address for monitor memory at 0xD0000. Also, I chose to position the monitor data beginning at address 0x1000. The application code follows the monitor code.

Application data space begins at address 0x8000. The ETS Monitor automatically unpacks global variables from ROM and places them in RAM, so the kernel has room to maneuver. Because our target has a BIOS, the target RAM size is retrieved via a BIOS call.

Notice an area that allows us to build a RAM Loadable Application. If this box were checked, a third linker command file is generated to facilitate this action. We won't do that right now.

That's it! I'll save the file as CIRCELL.VSB, so there are three new files in the working directory:

- CIRCELL.VSB-VSB data file
- CIRCELL.LKR-linker file for the ETS Monitor
- CIRCELL.LNK—linker file for the application

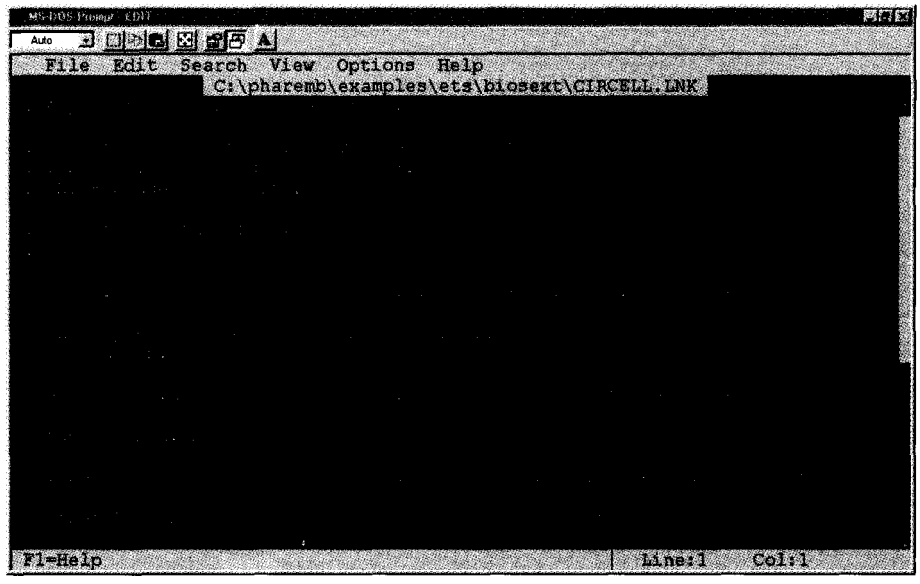


Photo 4: This linker command file uses the results of the ***LinkLoc*** operation against the *CIRCELL.LKR* file to build the downloadable memory image.

Understanding what's in each linker command file is the key to understanding the ETS replaceable-module system. CIRCELL.LKR is too long to list here, so I'll pull out the important parts. What I want to show is that all of the selections made within the VSB application are reflected in the CIRCELL.LKR and CIRCELL.LNK files.

The Monitor linker command file, CIRCELL.LKR, begins by specifying other linker command files to include. The .emb files are identical to the .LKR files generated with VSB.

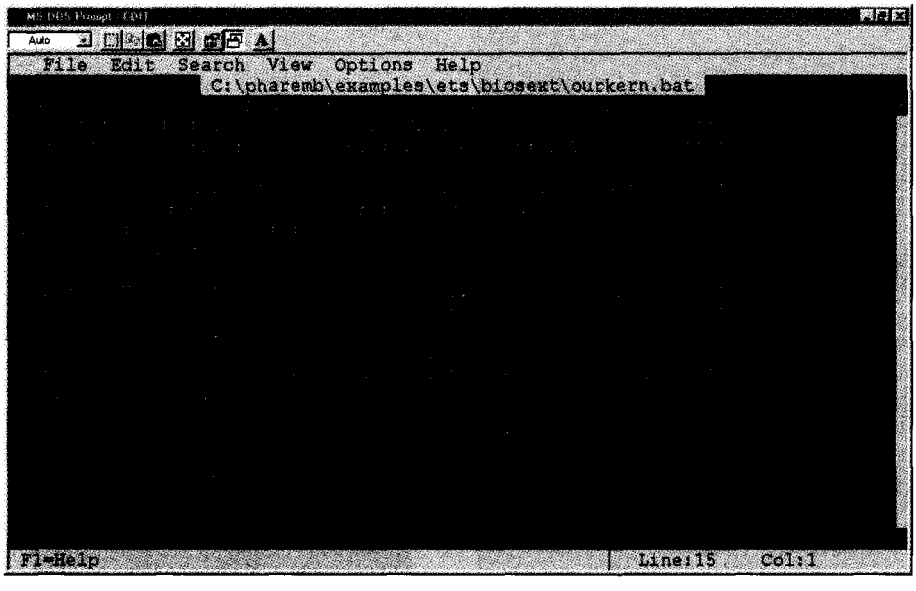


Photo 5: Once the environment is set and ***LinkLoc*** *has done its thing, the rest is automatic.*

Within the VSB property sheets, I specified that I required host communications support. Photo 3 shows that a supporting linker command file hostcomm.emb links into the ETS Monitor.

I also told VSB that the target couldn't support a diskette drive. A replaceable module in the form of a linker command file called noappds k.emb ensures our intentions are fulfilled. A closer look at Photo 3 reveals my soon-to-be monitor's name (BIOSMON.EXE) as well as the PC/AT support requested on the Drivers property sheet.

The CIRCELL.LNK file depicted in Photo 4 almost speaks for itself. Within the confines of this file is a linker command file that specifies how to link the application versus the compiler selected via VSB (@vcbase.emb).

Going down a little further, I see my PC/AT timer functions being linked in from the timer library. This line is followed by a directive to name the file CIRCELL.HEX and thus put it in Intel hex format.

This change is made because the actual final application file is in PE format. I'll download a memory image of the PE-formatted file to ROM with the "real mode" stub BIOSMON.EXE that's generated when I perform the final link.

The rest is logical and follows the data I fed the VSB application. That's what the ETS replaceable-module architecture is all about. To perform this same task for a different embedded target, the ETS system, under your control of VSB, simply plugs in

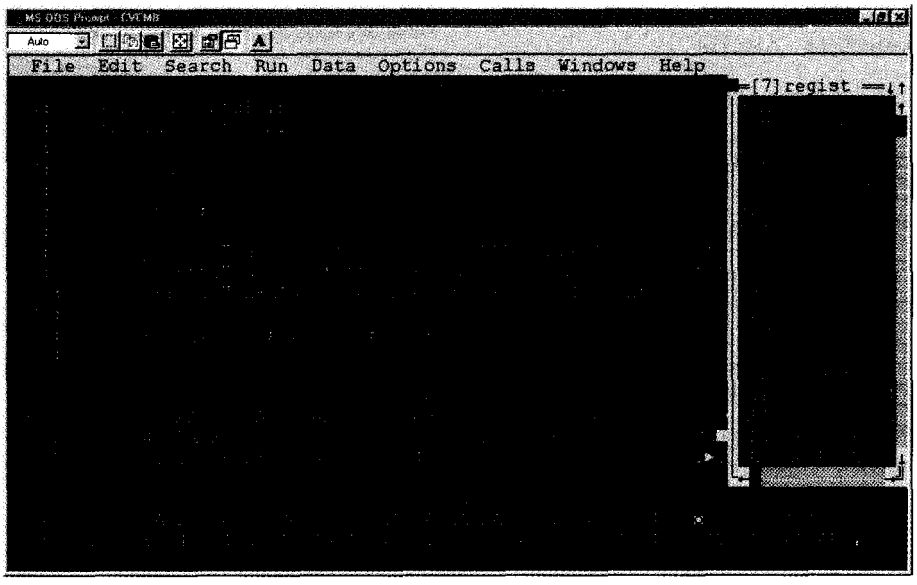


Photo 6: The dynamite is in place and the fuse is lit.

the correct modules from the appropriate linker command files.

PLACING THE DYNAMITE

Now that you've seen the hows and whys of the linker command files, you should have a good idea what's about to happen. The next step is to create a make file to generate our application. I took the liberty to do this while you were in the bathroom.

My handiwork is shown in Photo 5. This simple batch file uses the Visual C++ compiler to compile CIRCELL.C, resulting in the creation of CIRCELL.OBJ. The second command line links the ETS Monitor using the CIRCELL.LKR linker command file I created with VSB.

As a result, two files-the monitor BIOSMON.EXE and BIOSMON.MAP-are generated. The last command line links the application using the CIRCELL.LNK and CIRCELL.OBJ files to make CIRCELL.HEX, the downloadable file.

CIRCELL.HEX contains the ETS Monitor and my application. Debugging information was also requested, and CIRCELL.SYM and CIRCELL.MAP files are also created in this process.

Checkpoint time! To generate an ETS Monitor and the resulting application, I did the following.

I was born. I produced a C application source file. I used VSB to generate linker command files for use by Linktoc. I created a simple batch file to invoke the C compiler and Linktoc functions.

LIGHT THE FUSE AND RUN!!

The only thing left to do is test-drive the application. Using the FLASHLDR utility included with the '386EX target, I down-

loaded the CIRCELL.HEX file to boot flash. Let's do a debug run first and see if VSB put all that stuff in the right place. At the command line, I enter cvembcircell.sym.

After a little magic, Photo 6 appears. In the memory1 area of the CodeView screen, note that the telltale "55 AA" pattern is located at 0xD0000, and it looks like our data at 0x8000 in memory2.

Take a look at the registers. You'll notice that they are initialized-a no-charge feature provided by the ETS Monitor.

SEE THOSE STONES SHATTER

Doing the article has been a blast. Phar tap's TNT ETS is a perfect companion for theseriousembedded-application programmer.

The inherent flexibility of the replaceable-module architecture makes ETS a must-have for any embedded-application programming. If that in itself is not enough, the documentation and technical support are without compromise.

The TNT ETS is another powerful development tool that supports my premise-it doesn't have to be complicated to be embedded. APC.EPC

Special **thanks to Marty** *Bakal* **and Jim Phillips at** *ETS* **Support for supplying the blasting caps. Also, thanks go out to** *Scott* **Cope/and** *at* **Grammar** *Engine for checking the fuse when it got short!*

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

Grammar Engine, *Promlce* ***User Manual,*** Westerville, OH, V.3.4, 1995.

Intel Corp., Intel *'386EX* Embedded Microprocessor User's Manual, Santa Clara, CA, V.272485002, 1996.

Intel Corp., *Intel EXPLR1* Embedded PC Evaluation *Platform* ***Board Manual,*** Santa Clara, CA, V.272775001, 1995.

Phar Lop Software, *ETS* Kernel User's Guide, Cambridge, MA, V.TNTETSKern-I-I 195, 1995.

Phar Lap Software, *LinkLoc* ***Reference Manual,*** Cambridge, MA, Third Edition, 1995.

Phar Lap Software, *CodeView* ***Reference Manual,*** Cambridge, MA, First Edition, 1993.

Phar Lap Software, ***Using the*** *Phar* Lop *Visual* ***System Builder,*** Cambridge, MA, Second Edition, 1995.

SOURCES

Forth-Systeme Module '386EX board
Grammar Engine, Inc.
921 Eastwind Dr., Ste. 122
Westerville, OH 4308 1
(6 14) 899-7878
Fax: (6 14) 899.7888
BBS: (614) 899.6230
support@gei.com
CompuServe: 7 112 1,1603

'386EX evaluation board, **EXPLR1**
Intel Corp.
P.O. Box 58119
Santa Clara, CA 95052-8 1 19
(602) 554-8080
Fax: (602) 554-7436
http://www.intel.com/

Visual C++ Compiler **V.4.0**
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
(206) 882-8080
Fax: (206) 936-7329

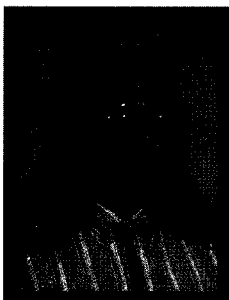
TNT Embedded **ToolSuite**
Phar Lop Software, Inc.
60 Aberdeen Ave.
Cambridge, MA 02138
(617) 661.1510
Fax: (6 17) 876.2972
BBS: (617) 661.1009
Telnet: bbs.pharlap.com

IRS

425 Very Useful
426 Moderately Useful
427 Not Useful

FROM THE BENCH

Part 2: PCB Layout—Time to Place the Donuts



Ever the designer, Jeff gives a helpful walkthrough of PCB layout software. You'll quickly see that modern software has brought us a long way from the old days of hand sketching and taping.

Jeff Bachiochi

I needed a wheelbarrow. The amount of brush I had to move was more than I was willing to carry by hand.

I started with the downed branches. Some large limbs required a chainsaw. Finally, clear of most of the movable obstacles. I brought in the mower to cut down the tall grass and small brush.

Having the right tools enabled me to clean up this eyesore in one weekend. I couldn't have managed without them.

This month, I'll discuss tools of a different sort-the PCB tools available to us thanks to the spread of PCs. Although many high-powered programs cost thousands, lower-cost PCB packages are well within everyone's reach.

Like the schematic-capture packages I discussed last month, PCB layout software has three distinct pieces-the parts library, the layout editor, and the postprocessor. Layout editors vary the most among manufacturers because they may include some high-powered and expensive options like autorouting.

OF TAPE AND DONUTS

After hours of sketching parts' placements, I'd fit the necessary boundaries of the proposed PCB outline. The frustration level was high as unworkable layouts were crumbled and tossed.

I try to remember the good side—once the layout worked on paper, it was easy. I just had to follow the sketch to lay down a little tape and a few donuts.

For the final phase, I cleared off the light table (a backlit drafting table) and placed a gridded mylar sheet on top to keep the tape on grid.

I worked on a 2x artwork. The donuts or pads were selected by their outside diameters. The pad's diameter was based on the part's lead, associated hole size, and leaving sufficient meat around the hole so the part had adequate pad area for a solder joint.

Pads for each pin of every part stuck onto the mylar. I checked each part's footprint for interference between parts when the PCB was assembled.

Since all parts were two times real size, it was easy to place the pads with a high degree of accuracy. Once all the parts' pads were placed on the mylar, the taper began laying down interconnections. He chose the appropriate width 2x tape and physically connected the pads according to the layout sketch.

Most PCB boards were single sided, so it made sense to spend a lot of time on parts placement since the traces could not cross one another. When a component's pad on one side of a trace needed to get to a component on the other side [see Figure 1], a jumper or O-R resistor bridged the existing trace.

This added parts and labor cost. If one part could be repositioned to serve as the jumper, it saved money. With double-sided and multilayer boards, this process became less critical.

LAYER IT ON

It was common practice to use multiple sheets of mylar for the different layers of artwork-the silkscreen outlined all the physical parts, a drill pattern held all the through-hole pads (e.g., one drill layer per pair of routing layers), and routing (from one to many, usually in pairs). Each routing layer carried connections either vertically or horizontally, and a change in direction (i.e., a corner) used an arc or a 45° angle to reduce signal reflections.

I've known some incredibly fast tapers. Laying tape in layers reminds me of how cartoons are made. The background and foreground are painted on separate layers, so they can move independently.

A 2x artwork was reduced by photographic means to 1x positives. This reduced alignment and registration errors by 50%. Can you imagine trying to place a 1x 50-mil pad and 8-mil trace with any accuracy? Next, films were sent to a PCB manufacturer for prototypes and/or production boards.

ENTER CAD

Years ago, I was embarrassed when the only way I could show off a good computer was to use the latest word processor or game. Today's CAD tools change this. They use computing power to the max. They're far from perfect, but they save time and money.

PCB layout process hasn't changed, but the tools sure have. Let's look at the basics for designing a layout using a CAD program as a parallel to the hand-sketched and -taped artwork.

PARTS LIBRARY

The parts library is by far the most important part of the PCB layout package. A schematic-capture package with complementary PCB layout tools has many links that define how a schematic part translates into a footprint part on a new PCB layout.

The postprocessing of the schematic drawing prepares a netlist file for the PCB layout program. Here's where you must make sure your two programs are compatible.

Don't assume all the translations are handled perfectly. Pay attention to the way the schematic part's pin is numbered and how it relates to the PCB part's footprint. E, B, and C of a transistor are not always 1, 2, and 3 on the physical part. For example, the 78-series positive and negative regulators are physically pinned out differently (see Figure 2), even though they can use the same footprint.

Like schematic libraries, parts libraries come well-stocked with the basic parts needed for any simple PCB. More expensive packages have extensive libraries from many of the leading parts manufacturers.

However, you always have to build special parts. The parts library can be as simple or complex as you need.

Full BOM (bill of materials) information can be stored with each part name if you wish to create a PCB without first using a schematic-capture program (PCB without a netlist) and still track the parts. On-the-fly change, add, and delete commands let you create a PCB with parts and connections from scratch.

Using separate mylar layers to hold different information on the old hand-taped artworks was the key to creating successful CAD artworks. A typical double-sided PCB for through-hole parts (not SMT) may have four layers—a silkscreen, soldermask (used for both top and bottom masks), top routing, and bottom routing. Once each layer is defined, the CAD program places the right data with its associated layer.

The decal or footprint for each part is defined by creating separate entities (i.e., lines, pads, and text) and placing them on predefined layers. When I define a new part's decal, I start with the part's pad definition.

Measure the part's dimensions to get the physical pin-to-pin relationships. The pin's cross section gives the necessary hole size and outer pad diameter. Give the hole size -10 mils beyond the actual pin measurement. This extra space depends on the fit you want the assembled parts to have.

If you have a spec sheet on the part, use the suggested hole size. Also check the pin-size tolerance, so you won't be caught when the next batch of parts comes in with slightly thicker pins.

Pads can be many different shapes, but the most common are round or square. Each layer-top, bottom, and inner (if needed for multilayer)—must have a pad definition.

Pads for surface-mount parts are unique. They're normally rectangular and are defined for only the top layer since they don't have through holes.

Unlike a through-hole PCB, where the solder pads are identical on both sides, the SMT PCB has a separate soldermask for the top and bottom layers. The soldermask prevents solder bridging by covering all but the solder pads with a nonstick mask.

The thickness of the donut or pad around the hole should be a minimum of -10 mils or the same width as the traces I'm connecting to it. Power traces require a wider pad to reduce the drop across it and maintain a higher mechanical bond (heavier parts).

After sizing and placing the pads in the footprint (make sure they're numbered the same as the schematic), draw the silkscreen decal on the schematic layer along with any text (e.g., mark pin 1 with a 1 or a dot, or a + for the polarity of a capacitor).

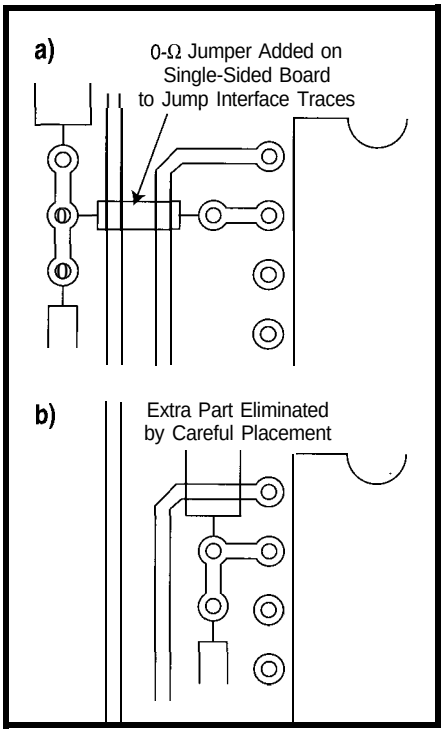


Figure I-Sing/e-sided *boards need additional thought to eliminate unwanted components.*

Since parts like resistors come in different sizes, the resistor part name can be linked to a number of alternate part decals. The netlist doesn't include data on the physical size of a part—only how it's connected to other parts.

So, the PCB designer must pay attention to the schematic notations to determine the correct part decal. Once all the parts are defined in the library, continue on with parts positioning.

PCB LAYOUT

Most designs have a predefined PCB size. You have to fit all ten pounds inside the proverbial five-pound sack.

You begin the second phase by defining the outside board dimensions and any necessary mounting holes. The layout editor simplifies this job. Draw the boundaries, and then cut, push, and pull them until you've got the right size and shape.

It's helpful to have a laser printer. Print out a lx artwork of the board outline and mounting holes. Cut out the board and place it in your enclosure so you can check the actual fit and location of the mounting holes. Any errors seen here are much easier to fix before any traces are laid.

Next, read in the netlist created by the schematic-capture package. This

list pulls parts from your parts libraries and places them one on top of another at the origin (i.e., 0,0 grid intersection] of your PCB outline.

Some PCB layout packages have an autoplace module that spreads the parts out on a predetermined x-y grid. Although it's useful in totally digital designs where the parts are similar in size, most designs have parts of radically varying sizes. I prefer to find the best placement (thank you very much).

When you move a part to a free location on the PCB, notice how connections to other parts stretch (rubberband). You can easily see where the shortest connections are by moving a part around on the PCB and paying attention to the rubberbanding.

Jockey the parts around until you find the best possible placement. It's a whole lot easier than sketching layouts with a paper and pencil.

Place decoupling capacitors on the ends of all the ICs to keep trace lengths as short as possible. Keep input and feedback components as close to amplifiers' inputs as possible. Keep components for analog and digital circuitry separate.

Once you're happy with the parts layout, print out another copy to check connector placement and areas where component height might be a problem.

AUTOROUTE

Every manufacturer likes to brag about their autorouter. I rarely use them because they slap down traces without letting me critique. If it does a dumb thing, I don't get to slap its hand.

Sure, I can clear the slate, alter the parameter it used, and rerun it. But, you can spend a lot of time fine-tuning the rules and rerunning a job. And, it isn't always clear why the router made the choices it did. It's usually a combination of factors.

Most of my designs are fewer than 50 parts on a board less than 25 $in.^2$, so I like to route the connections myself.

I start with the power and ground connections. When a design has more than two layers, a set of layers can be used as power and ground planes.

This technique has two advantages. First, it removes the burden of wasting precious routing space on power and ground. Secondly, the planes act as a capacitor, reducing the need for bypass capacitors on every IC.

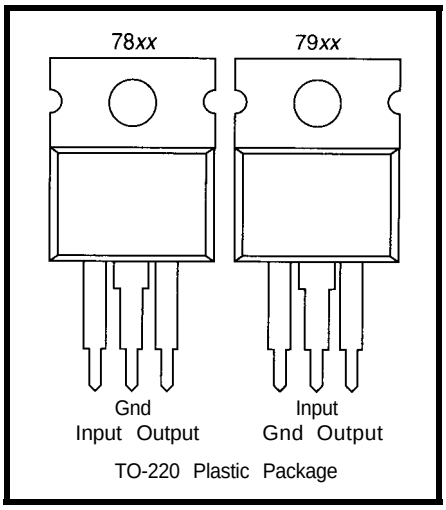


Figure 2-The same footprint may be used here. However, the pin numbers must match those in the schematic-created netlist.

The biggest disadvantage is the extra cost for the PCB. Making a multilayer (i.e., >2 layers) board is like making multiple double-sided boards and gluing them together. But, that's a whole other story.

Without power and ground planes, the power and ground traces (normally wider than signal traces) need to be placed first. Often, with minor repositioning of some parts, I can lay in a fairly straight gridwork.

Use a part's pin to change layers if possible. This reduces the number of vias (i.e., unsupported holes). Although a via isn't a problem, it's one more hole to drill and it all adds up.

Keeping horizontal routes on one layer and vertical routes on another increases the usable area per layer. But, no matter how you route it, there are times you can eliminate vias by running counter to the established direction for that layer. This is OK unless it cuts off a major throughway needed by other connections.

Some layout packages include interactive routing, which I think is more important than autorouting. With interactive routing, you can drag a connection through the area you want and the route is automatically placed. Push-and-shove routing forces preplaced routes off to the side, so you can squeeze a new route right through the middle.

If all this sounds easy, it can be. However, your machine's speed and available RAM determine how well these functions operate. Be sure to try out these important features before you plunk down any cash.

And, prepare for a lot of mouse clicking. Every function entails endless pointing and clicking. Most packages have special key combinations that replace clicking through pull-down menus.

One thing Windows has accomplished (good or bad) is that the programs are becoming similar in look and feel. These days, it's clever programming which sets off one package from another-not their looks.

Once fully routed, pay attention to the silkscreen layer. The part designations are usually every which way, depending on the placement rotation.

Choose an orientation for the PCB, and rotate or move all the designations so they are near their respective components and in the correct orientation. Avoid placing any silkscreen information on top of a hole or via because the ink disappears into the hole and can cause solder problems.

Documentation you may wish to include might be the board name or stock number, company name, and revision number. It's also a good idea to include some kind of layer designation on routing layers (e.g., component side, solder side, ground plane, power place, or just a number).

If you can afford the space, leave room for labeling jumpers. Jumper documentation is greatly appreciated by those who come after you. I hate looking up jumper configurations!

POSTPROCESSING

If we'd actually hand-taped a circuit, at this point, we'd need to visit the photo house to get lx positive artworks for fabricating the board. With the CAD software, we can deliver or modem the files to them for photoploting.

In fact, in many cases today, this step is skipped entirely. Fab houses can take the photoplot files and make the product without films-which can save big dollars. Postprocessing prepares these files and more.

Once routing has taken place, test your masterpiece. Although autorouting never (ha ha) disobeys any rules and creates errors, we humans might introduce a mistake here or there. So, it's wise to do a few checks.

First, run a connectivity check that compares your layout to the input netlist files and points out any discrepancies (e.g., an unrouted connection or a misconnected pin).

The second test I run is a spacing violation check, which can get confusing since there are rules for minimum spacing between everything (e.g., holes, pads, traces, corners, lines, text, board outline, etc.).

Since netlists don't carry information on parts placement and special connectivity concerns, here's where you don't want to leave things up to a machine. Sometimes, just connecting parts in the same net isn't good enough. They need to connect in a particular fashion (e.g., controlled impedance or multiple grounds).

Keeping analog and digital grounds separate is a must. However, they do need to be connected together at one point. To the CAD program, they are either connected or not. If you say they are, then the autorouter connects them anywhere, and the chances of a good choice here are nil to slight.

I add a small two-pin connector on the original schematic in which the grounds are connected one to each side. When the part is placed, I can easily control where the connection is made. I can place a small copper trace between the two pins or leave them to be connected physically by a shorting jumper placed on the pins later. The copper shows up as an error you can choose to disregard.

CAM (computer aided manufacturing) tools are the bridge to fabrication. The industry standard here is the Gerber photoplot file structure. Essentially, it describes how to move an x-y head and apply a tool-in our case, to open and close specially shaped apertures between a laser and the work surface.

The files are created by selecting items on various layers of the PCB layout. The selected items (or more accurately, how to draw the selected items) are all included in the file.

It's wise to also build print files of the same items. A printed picture is worth a thousand bad PCBs, if you get my drift.

There are Gerber photoplot viewers and editors available for checking the actual photoplot files. I find this prudent insurance against a bad file, but with a bit of confidence, a print file should do the job.

Along with each photoplot file is an associated aperture report file-a list of the aperture sizes and shapes necessary to plot the photoplot file. Each aperture is given a position number matching that in the photoplot file.

NOW WHAT?

Getting your circuitry produced in fiberglass is relatively easy. You might wish to use a PCB prototype house to produce a few PCBs.

Prototypes often don't have silkscreen or soldermask and are intended to be used for testing. However, today's prototyping houses get you excellent quality at very reasonable cost.

It's well within everyone's means to get a real board made instead of handwiring one or two boards. Check the back pages of your favorite magazine (hint) or other electronic periodicals for some contacts.

I've barely scratched the surface on creating a PCB layout using CAD tools. But, I hope I have taken a bit of the mumbo jumbo out of using CAD software and have enticed you into trying it out. For a source list of PCB layout tools, see last month's column.

If you want me to expand on any of these areas or even give an overview of what happens at the fab house, drop me a line. Your ideas are always appreciated. ❑

Jeff Bachiochi (pronounced "BAH-key-AH-key") is an electrical engineer on Circuit Cellar INKS *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

Motorola Lights ColdFire

SILICON UPDATE

Tom Cantrell

It has the reduced instruction set of RISC but with better code density and improved cache and debug capabilities. It's in the gap between RISC and CISC. But, don't simply dismiss it as a madeover CISC or embrace it as a RISC.

Keeping up with developments on the embedded RISC front is a full-time job, what with dozens of architectures and suppliers competing for attention. Things happen quickly, making unpleasant surprises for the unwary.

Only a year ago, the AMD 29k, enjoying great success in the laser-printer market and backed with a broad spectrum of chips and tools, resided near the top of the design-in contenders list. Suddenly, AMD decided to "Dr. Jack" the apparently suffering (from limited internal design resources given the 'x86 focus) CPU. Overnight, the 29k went from riches to rags, no doubt widowing more than a few system designs along the way.

Other RISC suitors mourned the untimely demise of their fallen comrade-for about a nanosecond. Hey, life is for the living and who cares if it isn't good form to ask the widow for a date at the funeral.

Those chasing the 29k estate had better keep one eye over their shoulder, though. A new and formidable suitor has come a-calling.

NEW KID IN TOWN

Actually, Motorola has been making ColdFire CPUs for some time, working on a custom basis with premier printer customer HP. (In fact, HP has licensed the core for internal use.)

What's new is-coincidental with the latest internal Motorola reorganization—ColdFire is taking its place in the 32-bit lineup squarely between the 68k and PowerPC.

The need for a product to fill the 'ISC gap is driven by the emergence of low price ($10–20), consumer- and controller-oriented 32-bit RISCs like the ARM, Hitachi 'SH, MIPS-derivatives from LSI Logic, IDT, and others. These chips are up to the task of competing with the aging 68k portfolio.

Meanwhile, bet-the-farm stakes on the desktop are propelling the PowerPC inexorably toward performance at any price.

The result is a widening and potentially threatening gap between the commodity CISC "controllers" and expensive RISC "computers."

Be warned that the arrival of ColdFire in the market and the need for Motorola to juggle three different architectures necessitates nimble (to put it politely) marketing. Motorola touts ColdFire as a "revolutionary RISC," while competitors claim it's neither revolutionary nor a RISC.

Ultimately, the datasheet specs, not PR spin, tell the whole story. So, check it out and decide for yourself.

C_ _

The ColdFire architecture certainly achieves a level of truth in advertising in calling itself a RISC. Strictly speaking, the instruction set is reduced, since it's a subset of the 68k.

Indeed, 68k users will be instantly familiar with the ColdFire architecture

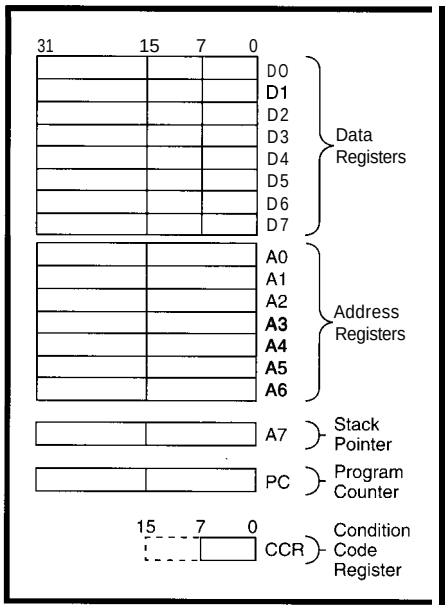


Figure 1—*The ColdFire programmer's mode/ is almost exactly the same as the 68k, composed of eight* data *and eight address registers. One difference is, although ColdFire retains fhe 68k user/supervisor privilege scheme, only one stack pointer (A7) is provided.*

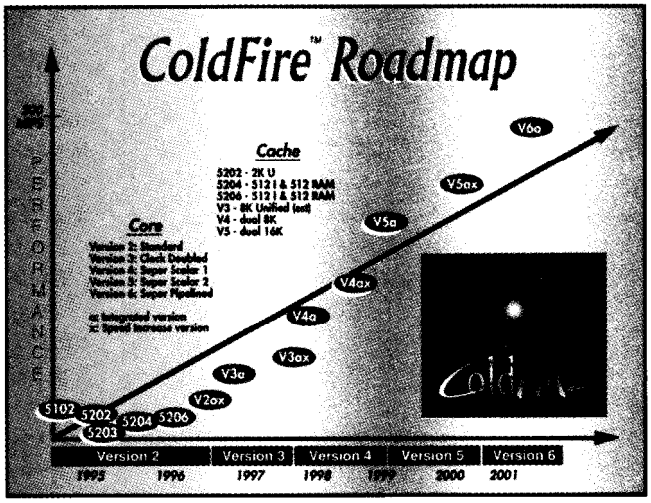


Figure *P-Though the first parts are focused af fhe low end ($10–20), Motorola plans to enhance ColdFire clock rate, architecture, and cache in the future.*

shown in Figure 1. Essentially, Motorola simply took out the more complicated and rarely-generated-by-compiler instructions, data types, and addressing modes. Switching from microcode to hardwire control with a four-stage pipeline speeds the streamlined instructions.

Though few may grieve over the passing of BCD and bitfield support, the trimming also includes stalwarts such as RO L / RO R (rotate) and the transistor hungry D I V (divide).

Despite the downsizing, competitors will likely point out that this 68k-on-a-diet meets few-if any-of the original technical requirements of RISC. For instance, RISC traditionally calls for lots (typically 32 or more) of registers, while Coldfire offers only the 68k complement of 16. Furthermore, RISC registers are supposed to be general purpose, while Coldfire, as the 68k before, splits the register set into address and data functions.

Traditionally, RISCs use the load-and-store concept in which instructions operate only on registers while memory is only accessed by load-and-store instructions. ColdFire axes load and store since instructions can operate directly on memory.

A final, though perhaps less controversial, departure from RISC dogma is the fact that ColdFire instructions can be one, two, or three 16-bit words. Compared to the traditional fixed 32-bit instructions of RISC, ColdFire achieves better code density, according to Motorola.

Motorola isn't the first to raise code density as an issue (the Hitachi 'SH and ARM Thumb come to mind). It doesn't matter much on the desktop, where it seems to take 16 MB to boot no matter what CPU you're using. However, in the cost-driven embedded world, code density may be a big deal.

I say "may" because the jury is—and should be-still out on this one. First, the truth of the matter must be decided, given the analysis is complicated by the usual benchmark and compiler caveats. Second, the overall impact must be assessed for a particular application.

For instance, the cost of memory continues to decline. Any code density difference might simply be hidden in ever-chunkier memory granularity. On the other hand, denser code squeezes more mileage from on-chip cache, achieving a higher hit rate per a given capacity.

CACHE ME IF YOU CAN

Speaking of cache, this is another area where conventional wisdom can and should be questioned. Once again, it's an issue where truth must be determined and the impact assessed.

The truth is that the degree to which cache has been dissected in the computer world is counterbalanced by a nearly complete lack of research on proper cache organization for embedded systems.

Should the cache be unified or separate for instructions and data? How much cache is required and, if separate, what's the instruction / data split? Direct-mapped or set-associative and how many sets? Line size? Write-back or write-through? Coherency?

At this point, I don't think anyone really knows the answers, so there's a lot of variation across the spectrum of embedded RISCs. Entry-level chips may include only a small (or no] cache, perhaps offering a small scratchpad RAM instead. High-end chips often feature the huge and complicated caches more typically associated with a computer than a controller.

More than the "right" instruction set, the best chip for a given application is likely the one with the "right" cache organization.

Such variability can be seen even within the ColdFire lineup as shown in Figure 2. The lean-and-mean (16-bit bus, 100 pin) 5204 relies on a simple 512-byte direct-mapped instruction cache combined with an instruction- or data-capable 5 12-byte RAM.

Meanwhile, the more powerful (32-bit bus, 160 pin) 5202 features a rather hefty 2 KB unified four-way set-associative nonblocking (i.e., "hit

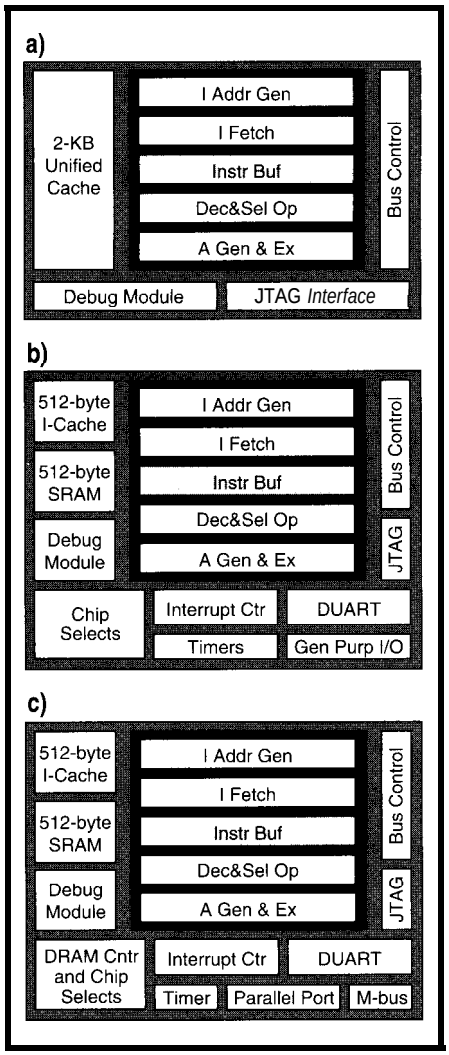


Figure *3-The* 5202 *(a) is the baseline Coldfire CPU, while fhe 5204 (b) and* 5206 *(c) cut cache size in favor of integrated I/O and sysfem logic.*

under miss") cache with a variety of programmable features, including write-back versus write-through, cache locking, and write buffer. As Figure 2 shows, ever larger caches are expected to accompany future architectural enhancements.

Taking a quick look at the chips announced so far, the 5102 is intended as a bridge for existing 68k customers. As such, it's provided with the otherwise-missing '040 instructions and addressing modes for complete binary compatibility with existing software. The idea is to get existing 68k code running on a 5102, establishing a solid hardware base for future software migration.

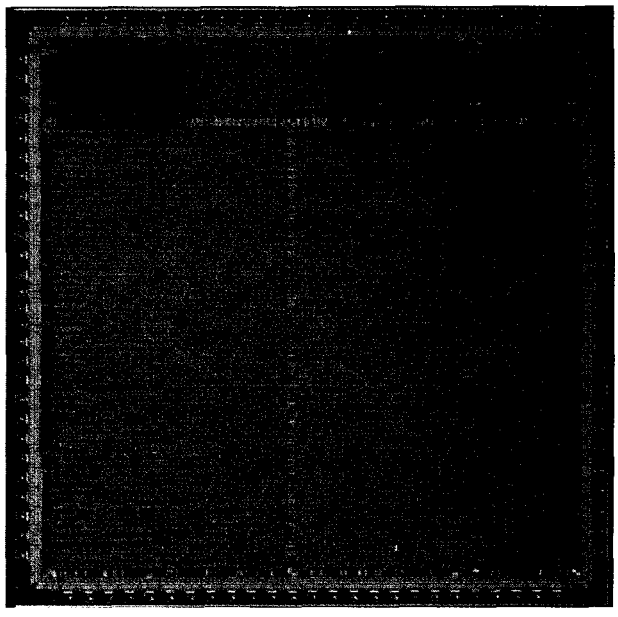


Photo 1--This *5204* die *photo proves that logic synthesis is* becoming as viable as *a mainstream design technique.*

Those prepared to abandon the past should skip directly to the 5202 which dispenses with the 68k baggage. As shown in Figure 3a, the 5202 combines the CPU with the previously described 2-KB unified cache, dynamic sizing (8, **16, 32** bit) bus controller, debug module, and JTAG interface.

If you prefer more I/O and less CPU, the 5204 shown in Figure 3b and Photo 1 combines the cost-reduced cache-plus-RAM setup mentioned earlier, a 16-bit bus, and a typical lineup of integrated I/O including timers, UARTs, and PIO, among other things.

In turn, the 5206 (see Figure 3c) builds on the 5204 with 32-bit data bus, glueless DRAM interface, and something called the M-Bus, which is apparently-I don't have any details yet-intended to accommodate some or all of the popular clock-serial interfaces like I²C, Microwire, and Motorola's own SPI.

Expect Motorola (and big customers) to easily and quickly spin new application-oriented ColdFires. Like the 68k before (and unlike the PowerPC), the ColdFire core can be integrated into Motorola's large library of well-proven peripheral functions.

PIN DE BUG DOWN

Systems-on-a-chip are a great idea, until debug time rolls around. There's no room between the transistors for a scope or analyzer, not to mention the cold coffee, stale munchies, and dog-eared copies of ***INK*** that litter my bench.

Actually, humor is probably the best way to cope with what's actually a pretty grim situation. Indeed, emulation and debugging problems have been embedded RISC's dirty little secret from day one.

Cache is one major culprit since the better it works, the less you can monitor activity from outside. In the extreme case (i.e., approaching 100% hit rate), the chip won't disclose anything about what's going on inside. Typically, the solution is to debug with the cache disabled, and then turn it on and hope that the timing changes and cache-management software aren't too hard to fix.

The problem gets worse with the inexorable growth of on-chip buses and functions. For instance, an on-chip DMAC could be performing an I/O transfer in parallel with all the other activity. The climbing number and speed of the signals to be tracked combined with the proliferation of chips and packages renders the historical bondout emulation-chip approach infeasible.

Recognizing the real-world challenges developers face, ColdFire goes further than perhaps any other CPU—RISC or CISC-in addressing the problem. It's to Motorola's credit that they devote significant die real estate to debug instead of more esoteric architectural features.

Taking a divide-and-conquer approach, ColdFire provides a variety of debug aids and regimes matching the various phases of hardware and software debug.

Current 68k users are likely familiar with the Background Debug Mode (BDM) scheme introduced on later members of that family. ColdFire carries the BDM concept forward with minor modifications.

Command	Mnemonic	Description	CPU Impact
Read A/D Reg.	RAREG/RDREG	Read selected address or data register and return result via serial interface	Halted
Write A/D Reg.	WAREG/WDREG	Data operand is written to specified address or data register	Halted
Read Memory Location	READ	Read sized data at memory location specified by the longword address	Steal
Write Memory Location	WRITE	Write operand data to memory location specified by the longword address	Steal
Dump Memory Block	DUMP	Use in conjunction with READ to dump large blocks of memory. An initial READ sets up the starting address of the block and retrieves the first result. Subsequent operands are retrieved with DUMP.	Steal
Fill Memory Block	FILL	Use in conjunction with WRITE to fill large blocks of memory An initial WRITE sets up the starting address of the block and supplies the first operand. Subsequent operands are written with FILL.	Steal
Resume Execution	GO	The pipeline is flushed and refilled before resuming instruction execution at the current PC	Halted
No Operation	NOP	Null command	Parallel
Read Control Reg.	RCREG	Read the system control register	Halted
Write Control Reg.	WCREG	Write the operand data to the system control register	Halted
Read Debug Reg.	RDMREG	Read the Debug Module register	Halted
Write Debug Reg.	WDMREG	Write the operand data to the Debug Module register	Halted

Table *I-Background Debug Mode (BDM) offers a set of basic commands that allow a development system fo establish control over the CPU. In the CPU Impact column, "Halted" means that the CPU must be ha/fed fo perform this command, "Steal" indicates that the command generates bus cycles which can be interleaved with CPU accesses, and "Parallel" means that the command is executed in parallel with CPU activity.*

Essentially, BDM provides the equivalent of a small ROM monitor in hardware with a special three-wire clock serial I/O port (instead of the usual UART) handling the host communication. The principle is that, relying on a "smart" host (i.e., PC or workstation), only the few simple commands shown in Table 1, such as read/write memory, read/write control register, and GO, are needed to establish complete control.

Implementation of the monitor function in hardware rather than CPU software means none of the target address space is consumed and the monitor and application needn't contend for shared resources like interrupts. Furthermore, unlike the typical software monitor, BDM allows a degree of overlap between application and debug operations. For instance, the debugger can read or write memory by stealing a few cycles from, rather than completely halting, the processor.

A software monitor's breakpoint function is typically limited to jamming a trap at the break address and thus is limited to catching instruction fetches from RAM. By contrast, ColdFire includes a full complement of hardware breakpoints including instruction address, data address, and data value.

Far from simple address comparators, the breakpoint scheme is quite complex and powerful. A wide variety of break conditions is configurable since qualifiers (e.g., address, transfer type, transfer size, etc.) can be masked and logically inverted (i.e., trap on an address or data within or outside a range).

The conditions are organized in programmable two-level hierarchy, allowing precise targeting of ornery code. For example, you can configure the debug module to break on the instruction fetch from address x that follows a data transfer at address *y*.

These are all well and good, but they don't solve the problem of tracing real-time activity in the cache. Motorola adds yet another resource in the form of 8 pins comprising 4 bits of processor status and 4 bits of debug data pumped out synchronous with the CPU clock.

PST[3:0]	Definition
0000	Continue execution
0001	Execute an instruction
0010	Reserved
0011	Entry into user mode
0100	Execute Pulse instruction
0101	Execute taken branch
0110	Reserved
0111	Execute RTE instruction
1000	Transfer 1-byte on ddata
1001	Transfer 2-byte on ddata
1010	Transfer 3-byte on ddata
1011	Transfer 4-byte on ddata
1100	Exception processing'
1101	Emulator-mode entry exception processing'
1110	Processor stops, waits for interrupt'
1111	Processor is halted'

'These encodings are asserted for multiple cycles.

Table 2—*For real-time trace with cache enabled, the 4-bit Processor Status (PST) bus tracks CPU activity every clock cycle. A separate 4-bit Debug Data (DDATA) bus outputs critical internal address and operand info.*

Obviously, you can't shovel 100+ MBps of internal bandwidth out a few pins. Here, reliance on a smart host provides the key, particularly in the sense that it has full and accurate knowledge of the application code.

Thus, there's no need to see every twiddling bit of internal address or opcode. Carried to the extreme, the host should be able to figure out what's going on knowing little more than an instruction is being fetched, a branch is being taken, or an exception is occurring. As shown in Table 2, these basics are easily encoded on the four status pins.

One big gotcha with the scheme is computed branches (e.g., a C case statement) in which the target is dynamically computed (and thus not known a priori by the host). Knowing the CPU took a branch doesn't help much if you don't know where it went!

Here, the data port comes into play, dumping the target address out in four-bit chunks. To prevent real-time bottlenecks, the connection between the CPU and data port is buffered with a two-entry (each up to 32 bits] FIFO.

The status and data ports are also made accessible to software debug schemes. The ColdFire Pulse instruction drives a predefined code ($4) onto

STATEMENT REQUIRED BY THE ACT OF AUGUST 12, 1970, TITLE 39, UNITED STATES CODE SHOWING THE OWNERSHIP, MANAGEMENT AND CIRCULATIONOFCIRCUITCELLARINK. THE COMPUTER APPLICATIONS JOURNAL, published monthly at 4 Park Street, Vernon, CT 06066. Annual subscription price is $21.95. The names and addresses of the Publisher, Editorial Director, and Editor-in-Chief are: Publisher, Daniel Rodrigues, 4 Park Street, Vernon, CT 06066; Editorial Director, Steven Ciarcia, 4 Park Street, Vernon, CT 06066; Editor-in-Chief, Kenneth Davidson, 4 Park Street, Vernon, CT 06066. The owner is: Circuit Cellar, Inc., Vernon, CT 06066. The names and addresses of stockholders holding one percent or more of the total amount of stock are: Steven Ciarcia, 4 Park Street, Vernon, CT 06066. The average number of copies of each issue during the preceding twelve months are: A) Total number of copies printed (net press run) 33,853: B) Paid Circulation (1) Sales through dealers and carriers, street vendors and counter sales: 4,615, (2) Mail subscriptions: 22,323; C) Total paid circulation: 26,938; D) Free distribution by mail (samples, complimentary and other free): 3,564; E) Free distribution outside the mail (carrier, or other means): 379; F) Total free distribution: 3,943; G) Total Distribution: 30,881; H) Copies not distributed: (1) Office use leftover, unaccounted, spoiled after printing: 564; (2) Returns from News Agents: 2,408; I) Total: 35,853. Percent paid and/or requested circulation: 87.2%. Actual number of copies of the single issue published nearest to tiling date are: (November 1996, Issue #76) A) Total number of copies printed (net press run) 32,000; B) Paid Circulation (1) Sales through dealers and carriers, street vendors and counter sales: 4,641, (2) Mail subscriptions: 22,126; C) Total paid circulation: 26,767; D) Free distribution by mail (samples, complimentary and other free): 2,009; E) Free distribution outside the mail (carrier, or other means): 500; F) Total free distribution: 2,509; G) Total Distribution: 29,276; H) Copies not distributed: (1) Office use leftover, unaccounted, spoiled after printing: 800; (2) Returns from News Agents: 1,924; I) Total: 32,000. Percent paid and/or requested circulation: 91.4%.
1 certify that the statements made by me above are correct and complete. Daniel J. Rodrigues, Publisher.

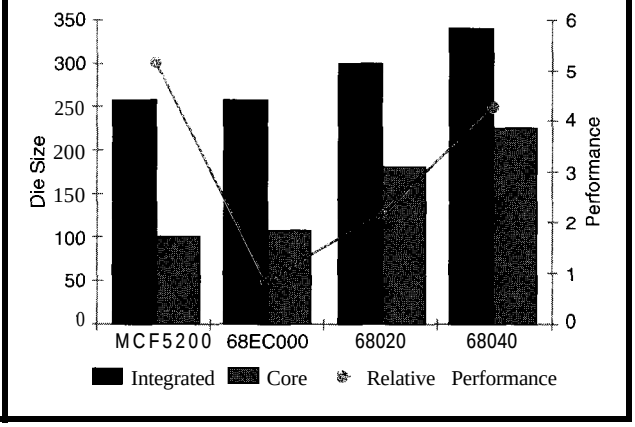


Figure 4—*Whether* it's a *RISC* or a *CISC, ColdFire appears to deliver excellent bang for the* buck *(i.e., MIPS/transistor).*

the status bus, useful for generating an external trigger signal. The **W D DATA (Write Debug Data)** instruction generates the same trigger and also funnels an 8-, **16-,** or 32-bit operand out the data port.

If that's not enough, ColdFire also includes a JTAG port. This IEEE standard defines a simple clock-serial-port mechanism to set or interrogate the level on a chip's pins. JTAG-enabled chips can be connected in a daisy chain allowing major portions of a board's functionality to be checked out without resorting to the "bed of nails" type testers.

Since JTAG is most useful during initial hardware checkout and production board test rather than ongoing development and debug, the JTAG and BDM clocked serial ports are multiplexed to the same pins.

RISC t CISC □ CRISC?

I'd fault Motorola more for calling ColdFire a RISC, except that the term has already been rendered meaningless by others. Step by step, each successive RISC has departed from the original minimalist dogma.

The fact that ColdFire dismisses not some or even most but in fact practically every tenet of RISC can simply be viewed as the culmination of the trend. Ironically, maybe the "most revolutionary" RISC is indeed a CISC.

The other reason a little exaggeration can be overlooked is that most savvy designers (i.e., *INK* readers) know that all the 'ISC hoopla has little to do with getting a real-world product out the door. Don't let the PR excess cloud your evaluation in either direction. Those who dismiss ColdFire as a made-over CISC are making as big a mistake as those who'd buy it just because it's labeled RISC.

The fact is, ColdFire seems like a very well-balanced and -positioned piece of work, not falling into the trap of ignoring the basics (like debug) in a frenzy to adopt the latest whizzy feature. At the same time, Figure 4 demonstrates that pragmatism doesn't necessarily make for a pokey CPU.

Just remember-price, performance, delivery, tools, support, and business relationships are ultimately far more important than architectural name-calling and petty religious wars. In the things that matter most, ColdFire packs a lot of power.

Tom Cantrell has been working on chip, board, and systems design and marketing in Silicon Valley for more than ten years. He may be reached by E-mail at tom.cantrell@circellar.com, by telephone at (510) 657-0264, or by fax at (510) 657-5441.

SOURCE

ColdFire
Motorola, Inc.
High Performance Embedded System Div.
6501 William Cannon Dr. W
Austin, TX 787358598
(512) 891-2000
Fax: (512) 891-2943
http://www.mot.com/hpesd/

I R S

431 Very Useful
432 Moderately Useful
433 Not Useful

CONNECTIME

conducted by Ken Davidson

The Circuit Cellar BBS
300/1200/2400/9600/14.4k bps
24 hours/7 days a week
(860) **871-1988—Four** incoming lines
Internet E-mail: **sysop@circellar.com**

The following message threads may be obtained in their entirety either from the BBS or on disk. See "Article Software" at the end of this column for more information.

Opto-coupler detector

The first thread this month starts with what looks like a simple question. The caller wants to build a circuit that generates a 50/50 square wave from an AC signal, using only an opto-coupler and some resistors. He's already built the circuit, but using proper current limiter resistors, the square wave has a duty cycle closer to 40/60.

People quickly come to the conclusion that it isn't possible using just resistors. After getting some good suggestions from others, the original caller mentions that the square wave's edges need to coincide with the zero crossings and the phases must match.

Several more very complete answers spell out a number of excellent solutions, but none produce a perfect 50% duty cycle. This is a very well reasoned thread that nicely demonstrates how a problem that seems simple on the surface can end up being quite complicated.

GPIB port to parallel printer

I'm sure you're aware of the old oxymoron "RS-232 standard." Just because a device supports RS-232 doesn't mean it's easy to connect to another RS-232 device.

The same can be said of GPIB. In the next thread, a caller wants to connect a Tektronix TDS520 oscilloscope with a GPIB interface to a dot matrix printer.

Other callers cite several sources of adapters, some questionable because they may no longer be in production or may not work in the application. This leads into some stories in which different devices equipped with GPIB interfaces couldn't talk to each other due to differences at the higher application level.

The final message sums it up nicely: ". .the problem is not so much in the raw communication. It is that the two ends still speak different 'languages."'

Press 'n' Peel

In the final summarized thread this month, we cover a topic that comes up every so often and that Jeff has done a "From the Bench" column about-making PC boards with a laser printer.

This time, the caller asks about a product called Press 'n' Peel. He's having a bit of trouble with it, but several other callers chime in to provide some hints for producing decent-quality prototype PC boards. It's certainly not for production use, but when you need a quick board to test, it looks like a good solution.

Home security systems

Msg#:10919
From: Tom Nickel To: All Users

After reading the many articles on home control systems, you'd think I would have applied some of the technology. But....

My home was recently broken into. Whoever it was took a couple VCRs, some tape decks, musical instrument amp, and so on. The worst loss was my vintage Les Paul Deluxe guitar-that's going to be hard to replace.

So now, this light comes on in my dimly lit brain. "Hey!" my brain says, "you need some nifty home security system stuff for sure!" Of course, I've replaced the broken window in my front door and installed "double-barrel" deadbolts in the front and back, as well as secured all the windows.

But, I think it would be really neat to have some sort of ultrasonic detector at the front and rear entrances. Maybe a speaker at both locations which announces a digitized warning that a silent alarm will sound. Or, a microcontroller that keeps track of all the "events" of proximity during the day. Maybe even a solid-state camera connected to a video recorder hiding in the house somewhere? At any rate, has anyone had experience with what techniques are most effective (and of course, fun to build and use)?

Msg#:11104
From: Henry **Schaper** To: Tom Nickel

The best home security system is one that keeps the burglar away from your home all together. In my case, I use an X-10 system to turn lights on, off, brighten, and dim as if someone was at home. Every house around me has been robbed in the past 15 years I've used it. Some several times. The best feature of using the X-10 system or any home control system is that it uses the exact same lights you normally use, not some table lamp that is only on when you're away.

Make it a challenge to figure out if anyone is home. Midnight Movers Inc. is not in business to work hard or to think.

A home control system can also give you the needed central controller for an alarm system in case Midnight Movers Inc. decides to pay you a visit.

CONNECTIME

Msg#:11259
From: Tom Nickel To: Henry Schaper

Thanks for the advice. But, here's another thought. I was talking with a person who sells security systems. He told me that the majority of residential break-ins are during the day! And through the front door of the house. That's what happened to me-broad daylight, clear sky, nice weather—in the middle of the afternoon, through the front door. So, the "turn-on-the-lamp" technique would probably not have helped during that time of day.

Msg#:11565
From: Henry Schaper To: Tom Nickel

Good point. It is important to have a good strong front and back door properly hung to at least slow up the uninvited daytime mover. It doesn't hurt to have the landscaping so it leaves a clear view of all the doors and windows. Perhaps the cheapest thing is alarm warning stickers, at outside eye level, on all the windows and doors. You can get them from Radio Shack. The idea is to always make your house look harder to get into than your neighbor's.

The better part of my town is having a problem with thieves cruising in vans. When they find an open garage, one person runs into it and takes whatever is handy and of value. They run it back to the van and quickly drive off. They hardly even stop. The poor homeowner returns his mower from mowing the backyard to find his mountain bike is gone.

Msg#: 11643
From: George Novacek To: Tom Nickel

Years ago, I was in the security business. There are quite a few books and articles on the subject. In terms of intruder detection, the most popular are passive infrared detectors (PIRs). Ultrasonics went out many years ago. Microwaves never gained much popularity. In addition, you need door and window magnetic switches. All these devices can be purchased. The only area where you might have some fun is the controller.

The major issue with alarm systems is what do you do having detected an intruder? At one point, local alarms and sirens were the most popular solution. But with the overabundance of false alarms, they became a nuisance and are often just ignored. The false alarms are another good reason why you should not be building your own detection devices, unless you want to antagonize your neighbors and the local cops.

Today, the majority of systems use a telephone interface to monitoring stations which register and verify the alarm and dispatch help. Similar to the cell-phone market, money is in communications, not the equipment. If you look around, you will find that some alarm companies will install a system in your house for free or a small fee ($100–200) if you sign up for a period of time for monitoring, at about $20 per month.

It is doubtful any alarm company will monitor your home-built system, as they are often liable for fines in case of false dispatches of police. An unmonitored system is just as useful as those systems many years ago which played a tape with a barking dog.

Msg#:11644
From: George Novacek To: Henry Schaper

For the most part, North American homes are a security joke. You will see homes with strong solid steel doors flanked by lights or containing windows which take nothing more than a bunch of keys to shatter. The backs of homes are full of patio doors with great locks and bolts. The glass needs a simple kick-and don't worry about the noise of shattering glass. Some jurisdictions require that the glass will not shatter for safety reasons (remember, Big Brother must protect us and the poor criminal from getting cut). And, there are simple and quick methods to overcome this problem.

I had a salesman who used to show proud owners of (commercially available) security doors how easy it was to get in. It never took him more time than if he had the key. You'd be surprised how easy it is to tame a vicious dog or how ridiculous some of the measures that homeowners take look when viewed by a professional thief.

Using alarm warning stickers is an old trick which could have been successful at the advent of home security systems. At that time, quite a few quick-buck artists jumped on the bandwagon and started producing those labels. Every self-respecting crook knows who the bona fide alarm companies in the area are. And their employees are strictly forbidden to give out the stickers unless you buy the system.

Msg#:11690
From: Henry Schaper To: George Novacek

It's all relative. I'm just interested in my house looking harder to get into than the competition-my neighbors' houses. Short of doing a house in poured concrete with no windows, you can't keep out a determined crook.

Using stickers is an old trick that will not slow down a self-respecting crook. That's fine. However, I would like the beginners to learn their trade at someone else's house.

The unfortunate thing about a monitored alarm system is that by the time the police get there, there is little the police can do. A self-respecting crook is gone with whatever he could get away within given the response time. All the $20 a month gets you is a crime report with the EXACT time of entry.

CONNECTIME

Msg#:11735
From: Tom Nickel To: George Novacek

Thanks (everybody) for the responses. I have been learning a lot since the incident at my house. I talked with the crime-prevention division at my local precinct. They had a lot of good information.

One interesting point they made: an alarm system is one's third line of defense. The first being lighting and landscaping (i.e., making sure hedges are trimmed-mine are not-as well as trees, and that there's discouraging lighting at night).

The second defense is doors and windows. They recommended Lexan in any door window within 40″ from the door lock. Glass is the weak point in the home's security. Double cylinder deadbolts are discouraged as they "trade safety for security," but I used them anyway.

After doors come ground-level windows. A simple pin or nail can lock the bottom sash to the top sash if they're wood. Then basement windows.

Then come the security systems. And, statistically, a break-in is only 3–5 minutes in length, which is too short a time for the police to respond. Plus, in my city, one is fined for having more than two false security alarms per year.

Still, I'd like to install a camera, controller, and video recorder to see who's sneaking around the neighborhood during the day.

Msg#:13094
From: George Novacek To: Henry Schaper

You touched on some very contentious issues of home security.

1) There are two camps when it comes to alarm and its annunciation. One group claims that you want bells, sirens, flashing lights, and what have you to scare the burglar and get him running. The other camp claims that you want a silent alarm at a monitoring station, because the panicking burglar is liable to hurt someone or take a hostage, and it is more important for the cops to catch him than to scare him away. (I don't agree with that. The guy's out on the street before a report is filed.)

2) There is a similar division when it comes to alarm companies' stickers. Some claim that the would-be burglar wants no more trouble than absolutely necessary, so he will avoid premises with alarms. Others claim that the stickers act as a challenge. Both camps agree that short of the druggies and guys with the IQ of a rubber plant, phony stickers are tantamount to advertising that you have no protection at all.

Msg#:14524
From: Pellervo Kaskinen To: Tom Nickel

Have you already gotten the repeat visit?

When I was in Nashville, an acquaintance of mine was burglarized. The police advised him to get an alarm system in a hurry. But, he always did things "right" rather than rush. So, he was still looking for the best alarm system when, six days later, the same burglar paid a second visit.

The police and insurance companies informed him that this is more the pattern than the exception. The burglar learned something about the available "good stuff" on the first visit. Maybe even made arrangements to quickly deliver it to the pipeline. In any case, while the first time mainly caused some general damage, the second time there was a considerable loss of the more valuable stuff.

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.

Message threads summarized at the beginning of the column are also available on the BBS for at least six months after they are first posted. The subject line at the start of each summary matches the subject used on the messages themselves. Simply call the BBS and search for those subject lines to find the message threads.

For those unable to download files or messages, 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, P.O. Box 772, Vernon, CT 06066, or use your Visa or Mastercard and call (860) 8752199. Be sure to specify the issue numbers with your order. Please add $3 for shipping outside the U.S.

IRS

434 Very Useful 435 Moderately Useful 436 Not Useful

PRIORITY INTERRUPT

No Nostradamus

Well, it's the end of the year, and I assume I'm supposed to say something prophetic, reminisce about the past year, and give my predictions about next year. Yeah, right.

First of all, you can forget about me being prophetic. If I was that smart, I'd have done something grandiose and be dealing with middle-aged retirement already. I'd be on a boat off the coast of Tortuga where the definition of "laptop" is a magazine and a mai tai. About the closest I ever get to prophetic is knowing that my odds for success are about the same as flipping a coin.

Unfortunately, it's December and certain things are expected from a technical magazine. One of the requisites is to talk about technical issues and forecast the future. Of course, a one-page editorial can fit only so much prognostication, so I'll be brief.

First and foremost for many these days is the Internet. I think it's a wonderful technical tool as well as a potent vehicle for social and scientific interaction. Its continued high-velocity expansion will follow the money trail-unfortunately. But as long as competitive services and sources don't become monopolistic, the growth can be enjoyed by all.

My greatest fear is government intervention. (You know, "Hello, I'm from the government, and I'm here to help you.") We need to be careful of knee-jerk reactions and regulative responses that ultimately destroy free speech for the majority. Government is no more a source of freedom than it is a source of revenue. Flame off.

Second is the PC industry itself. For many years, game manufacturers have been trying to produce the true "home computer" as an extension of the TV-based video game-one cartridge for games, another for word processing, yet another for check balancing, and so on. True desktop-PC manufacturers have been approaching the same goal from the opposite end through lower cost equipment, VGA-to-NTSC adapters, and an infinite variety of peripherals. In reality, I think the "real" home computer of the future (separate from the specific-function, business-oriented desktop) will be a networked or multiplexed entertainment device centering around an Internet connection and the television.

While it's conceivable that every TV will have an internal PC and be connected to the Internet, such redundancy is expensive. Instead, I think we'll have a low-cost Internet box, which also might include the functions of a traditional full PC for a little more money. Such a device will be wired or networked to all the TVs in the house and provide the user-friendly interface espoused by the video-game manufacturers but include the power and performance only an Internet-connected full PC can provide.

The closer association between TVs and PCs, combined with the potential for an eventual HDTV digital standard, will ultimately result in higher resolution TV displays. Displaying VGA on a TV today requires a VGA-to-NT% converter to lower the resolution,

While adequate for most activities like browsing and games, competitive pressures will ultimately force TV manufacturers to accommodate higher resolutions directly. At the same time, we'll see a plethora of new digital video-compression technology-both hardware and software. The goal of a TV/PC on a chip may, in fact, come to pass.

Finally, there's the issue of home control and building automation. While there may in fact be some advances in automated energy control and lighting in commercial sectors, I don't envision any drastic changes in the rate at which the average home is being automated. Home control is a business with few major participants and virtually no focus. Typically, the only justification most homeowners have for any reasonable-performance home-control system is that it invariably includes a high-performance security system. Commercial alarm manufacturers focus on alarms and liability. Home-control manufacturers focus on convenience and entertainment. No one has figured out how to combine a suitable level of convenience, justified by the proper level of security, and sell it in volume. Of course, we're still trying.

Well, there you have my opinion on things to come. If I hit everything right on the nose, I expect you'll rank me up there with Nostradamus. If I hit a big zero, just remember-I warned you ahead of time. In either case, I win.

Steve

steve.ciarcia@circellar.com