

...Recursion • Jon Bentley on Recursion • Jon Bentley on...

#286 JUNE 1998

Digital
Signatures
&
Microsoft's
CryptoAPI

Dr. Dobb's

JOURNAL

SOFTWARE
TOOLS

PROFESSIONAL
PROGRAMMER

<http://www.ddj.com>

PATTERNS & OBJECT-ORIENTED DESIGN

- Automatic Pattern Identification
- Java & the Visitor Pattern
- AntiPatterns
- Implementing Associations

Designing Class Libraries
for Visual Builders

Designing Technomadic Systems

\$4.95 (\$5.95 CANADA)



Inside Caller ID

CGI Programming & Visual Basic

Secure Communications
& Java Applets

A Miller Freeman Publication



Designing Technomadic Systems

Getting hardware devices to interact

Steven K. Roberts, N4RVE

My original motive was simple enough. Horrified by a view of the American Dream from the curtained windows of a three-bedroom ranch in suburbia, doing things I didn't enjoy anymore to pay for things I really didn't want, I hit System Reset. Six months later, in the fall of 1983, I put my house on the market and moved to a recumbent bicycle—I was a 30-year-old technomad heading across America with a primitive laptop, solar panel, and xnet connection.

The miles passed by the thousands, the adventures yielding a lifetime of memories, romances, and bizarre discoveries. Twice over the subsequent eight years, the seductive allure of technology inspired a complete system redesign, culminating in "Behemoth." This silicon-encrusted unix-cycle rolled out of a donated lab at Sun Microsystems in 1991 and lumbered down the highway with a Qualcomm satellite link to the Internet, a killer CD stereo, console Macintosh with handlebar chord keyboard and ultrasonic head mouse, PC with helmet-mounted display, repackaged SPARC with color LCD, 72 watts of solar panels, 105 speeds, robust ham radio station, audio/serial routing network, helmet cooling system, and full bicycle touring load. The whole rig totalled 580 pounds, not including a sweaty bio-engine.

My next technomadic move was to water, and here I am, deep into the fabrication of twin Microships—pedal/solar/sail canoe-based trimarans with enough em-

Steve is a freelance writer and engineer who can be contacted at wordy@qualcomm.com.

bedded systems in their pressurized consoles to make Behemoth look downright primitive.

Microship Overview

The Microship design concept is optimized for open-ended coastal and inland travel without dependence on marinas or other shore facilities. These matching boatlets (currently under construction in our Camano Island, Washington, laboratory) are folding multihulls with roller-furling freestanding sails, deployable recumbent pedal-drive units, and 480 watts of folding solar panels driving retractable thrusters. Each boat



(one for me, and the other for my partner Lisa) is capable of deploying lightweight wheels and being manually hauled out or launched without needing trailer and tow vehicle. They carry basic camping-scale life-support resources, including water filtration and desalinization, but we assume ready availability of shore facilities since the on-board stores inventory is small. (Our travel plans are primarily coastal and inland waterways, beginning with 12–15,000 miles in North America, then hitchhiking with the boats to Europe.)

Despite severe space and weight constraints, the Microships carry an extensive network of electronic systems, as well as

pressurized packaging to keep them alive in a corrosive environment (a nontrivial problem in itself). As with Behemoth, all systems can be loosely grouped into two overlapping categories—infrastructure and applications.

At the infrastructure level, the system has to deal with power management and distribution, thruster control, routing of audio/video/serial channels on demand, ship security, internal performance tracking and diagnostics, console pressure and temperature monitoring, dynamic network configuration (including remote wireless control panels), and so on. In short, this includes all the low-level tools that keep the boats working, provide access to resources, and are always on and must therefore use power as efficiently as possible. In this category, every milliwatt is dear.

The applications layer includes obvious nautical systems, such as navigation and communication, along with digital-video production and camera management, environmental data collection with hourly posting of telemetry blocks, Internet access, ham radio, manpack tracking, the whole suite of PDA and productivity tools accessible while sailing or off-boat, and ongoing software development.

It became obvious early in the development of the Microship system that some unique internal networking challenges were present. These included:

- Random software-controlled interconnection of analog and serial channels.
- Distribution of independent control resources across a low-level network to simplify development and minimize power drain.
- Integration of two autonomous boats into an apparent single system.
- Wired local and wireless remote graphic front end access via off-the-shelf PDAs.

Grand Central Station

One of the nastiest and most troublesome problems facing this design was the sheer number of hardware devices that have to

"A POWERFUL, FLEXIBLE LIFESAVER" - WINDOWS MAGAZINE

"A MULTITUDE OF HIGH-END FEATURES" - PC MAGAZINE

INSTANTLY SEARCH GIGABYTES OF TEXT



FIND ANYTHING

Text search options include:

- relevancy-ranked natural language
- thesaurus (for synonym or concept searching)
- fuzzy (adjustable from 1 to 10 for OCR and "spell1ng" errors)
- and/or/not
- proximity
- wildcard
- phonic
- field
- numeric range
- variable term weighting.

ANYWHERE

- Works with all popular file types.
- Displays retrieved files with hits highlighted.
- Built-in image viewer.

INSTANTLY

- Indexed search time is usually under a second.
- Also does unindexed, & "combination" searches.

■ dtSearch 5.0 for Win 95/NT (32-bit) \$199 ■ LAN 5 \$800 ■ No charge to evaluate.

ADD dtSearch's PROVEN TEXT RETRIEVAL ENGINE TO YOUR PRODUCT FOR THE PC, LAN OR INTERNET

- Unlimited capacity
- Sample source code for C/C++, Visual C++, Delphi, Visual Basic, ActiveX/ASP developers.
- Sample source code for a Web application.
- Can convert retrieved files to HTML for display with popular Web browsers, with the hits graphically marked.
- Displays PDF files in the Adobe Reader, with hit highlighting.
- \$999 for UNLIMITED concurrent use on a SINGLE Internet or network server.
- Royalty-free licenses available.

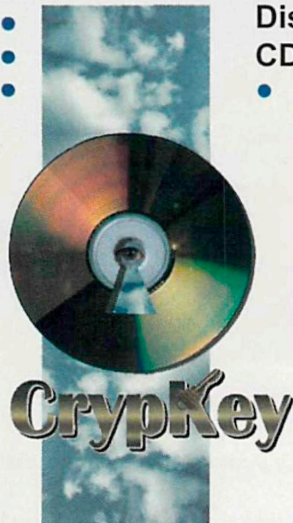
1-800-IT-FINDS ■ www.dtsearch.com

703-413-3670 ■ fax 703-413-3473 sales@dtsearch.com

"A SLEW OF SEARCH PROTOCOLS" - PC COMPUTING

AD LINK 369

Serious Copy Protection and Software Licensing with No Hardware Key or Disk Key



Distribute your software via Internet, CD or floppy with full security.

CrypKey provides solid copy protection with a broad range of features to control all aspects of your product's operation. You can offer free, automatic one-time trials, turn demos into full multi-user network versions, or sell specific options by phone, fax or email. Sell usage by time, runs, features - the control is all there - easy to implement, and even easier to support.

CrypKey also offers choices - integrate our SDK into your software, or use CrypKey Instant to protect your compiled exe in 5 minutes; no coding.

Try it free from our website to see why we're rated #1 by professional programmers who seek the best combination of security, features, ease of use, reliability & price.

100% Satisfaction Guaranteed!

<http://www.crypkey.com>

Kenonic Controls Ltd., Calgary, Canada Ph: 403-258-6200, Fx: 403-258-6201, email: info@crypkey.com

AD LINK 310

(continued from page 54)

interact. I'm not talking bus-compliant systems speaking a standard data format here. I mean cellphones, marine radios, video cameras, sensors spewing NMEA data-streams, frame grabbers, dedicated control interfaces and translators, and the like. From my bike experience, I know that any "complete set" of anticipated interconnects will become obsolete the first time a seductive new gadget arrives. (By the end of its era in 1988, the Winnebiko II was a study in reassigned console switches and multiple layers of annotation scrawled upon already poor documentation.)

For Behemoth, we developed an audio crossbar around the Mitel 8816 microcontroller, vastly simplifying those spur-of-the moment connections that are irresistibly associated with having lots of devices. It took only a few lines of Forth, for example, to respond to a latitude-longitude change without the right password by dialing 911 on the cellular phone and having the speech synthesizer intone, "I am a bicycle and I am being stolen. My present coordinates are..."

This sold me on the crossbar architecture for like-flavored signal paths. Consequently, the "Grand Central Station" region of the Microship system, see Figure 2(a), includes three crossbars (a cabling-intensive zone in the console with 244 connectors):

- The audio crossbar (Auxbar) allows up to eight simultaneous connections among any of 32 sources and 32 destinations. At the hardware level, it's a stack of two boards bristling with gold-plated RCA jacks, so all we have to do to add an audio device is run a cable and add a line to a table to allow reference by name. Lisa's boat joins the network via a short range stereo full-duplex transceiver for quality audio, and a 70-cm ham-radio link for voice-grade exchanges— devices that are, of course, merely addresses on the Auxbar. At Lisa's end, a miniature version of the same design allows the addition of multiple devices as well.
- The video crossbar (Vixbar) is similar, but optimized for higher-bandwidth signals and the need to channel switch during retrace intervals to prevent glitches. The system can handle eight simultaneous links among any of 16 sources and eight sinks, and, since it is less likely to be on call 24 hours a day, is supported by a dedicated Forth node and software-controlled power switch. This network is also expanded to the other boat via a video transceiver, since Lisa runs a Draco Casablanca video-editing system and needs to select camera channels on the fly. Like audio, this system appears as an array of RCA jacks and is trivial to expand or edit. One-line commands make or

break any link, add a receiver to an existing connection, or clear all of them.

- The serial crossbar (Sexbar) provides up to four simultaneous bidirectional connections among any of 32 serial channels—28 of which appear as a dense 4x7 array of DB-9 connectors pop-riveted together. This one is particularly amusing, since it also has to solve the annoying problem of RS-232 “polarity,” those eternally incorrect pins 2 and 3. The code in the Sexbar controller, whenever asked to establish a connection, first zips through the involved pins of both channels, connects them one at a time to a window comparator, sees who’s receiving and who’s transmitting, then assembles a virtual straight cable or a virtual null modem cable as needed. The process is so quick that we often link serial gadgets together just for the fun of it. (That’s a statement I haven’t been able to make since about 1973 when the process of making machines communicate was still intrinsically astonishing and UAR/Ts were radical new alternatives to doing it with shift registers and control logic.)

I occasionally take flak in these Internet-savvy days for using such primitive networking tools as vanilla serial channels loping along at a poky 9600 baud via

hardware-intensive crossbar. Why not just give everything a TCP/IP stack and do it right? Well, if the involved devices were all “real computers” that would be sensible, but most of the serial widgets we’re using are small, low-power, dedicated controllers—even PIC processors acting as translators between RS-232 and proprietary protocols like Sony LANC and Dallas MicroLAN. Sticking them all on Ethernet would be power-hungry overkill, in most cases adding far more communications hardware overhead than the devices themselves!

Finally, the power-control system deserves a comment here, even though it’s not a network, per se. In keeping with the spirit of maximum flexibility, all devices that are even remotely power hungry are hanging on solid-state relays, in turn wired to Phoenix Contact blocks that carry a few dozen bits of parallel I/O attached to the Hub processor. It is a trivial process (running one wire) to add a power-control channel with its own status LED, fuse, and pair of switched screw terminals; it takes just another one-line command to turn anything on either boat on or off.

With few exceptions, all random pieces of communication, entertainment, data collection, security, and control hardware appear to the Forth Hub as nothing more

than some subset of audio, video, serial, and power channel names.

A Multidrop Network of Multitasking Nodes

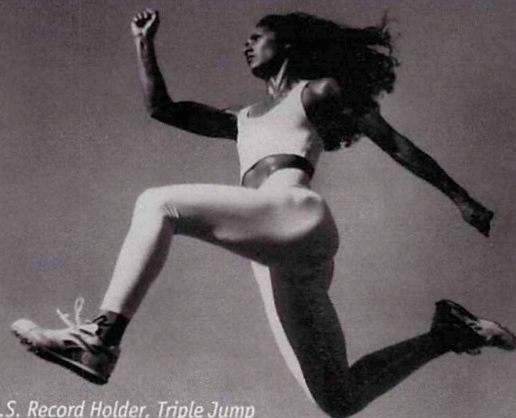
The number of autonomous applications on the boats is enough justification for a distributed collection of microcontrollers. Although one PC could certainly do everything with most of its clock cycles lying fallow and its cache tied behind its back, we’d gobble up precious amps while doing nothing but monitoring a few security sensor bits at anchor. From a power-management perspective, as well as development simplicity, the sensible approach is to deploy microcontrollers as needed.

If I were beginning the fabrication of this control network today, with almost all nodes in a single enclosure, I might take advantage of the completed Sexbar and tie the console ports of our New Micros (<http://www.newmicros.com/>) 68HC11 boards to numbered channels, then let the Hub make and break links to its auxiliary port during every node scan. But when we started back in 1994, the assumption was a much larger boat with a dozen nodes in multiple enclosures distributed along 30–50 feet of cable, so a multidrop network was a more sensible solution. Now, with all the

jump

into system development with the enterprise modeling tool champion.

In enterprise modeling you can't afford to be caught standing still—or using stand-alone products. Only the System Architect suite of tools allows you to jump from Business Processes to Objects to Data Modeling. Team collaboration and productivity are maximized through its multi-user networking and repository-based capabilities. Popkin has over 10 years of experience designing and producing software to meet your needs now, and far into the future. So stop standing still. Try us FREE for 30 days!



Sheila Hudson, U.S. Record Holder, Triple Jump



POPKIN
SOFTWARE

Enterprise Tools For Success

www.popkin.com/enterprise/540

FREE System Architect 30-Day Eval: 800.732.5227, x540

Argentina 54-1-328-0324 * Australia 61-3-6234-6499 * Benelux 31-30-466-55-30 * Brazil 55-11-5181-0246 * Chile 56-2-633-1733 * France 33-389-44476 * Israel 972-9-9-598-999 * Italy 39-49-8283411 * Korea 212-989-9809 * Malaysia 60-3-757-1806
Mexico 525-531-6768 * Spain 34-3-494-8200 * Sweden 46-8-626-8100 * Switzerland 41-61-6922666 * United Kingdom, Ireland, Denmark, Poland: 44-1926-450858. © 1998 Popkin Software & Systems. The System Architect logo is a trademark of Popkin Software & Systems. All other brand and product names are trademarks or registered trademarks of their respective holders. Popkin Software & Systems, 11 Park Place, New York, NY 10007. Phone: 212.571.3434. Fax: 212.571.3436

code well developed, integrated with the multitasker, and baked into EEPROM, there's no way we're changing—even if it does seem slightly redundant to have both multidrop and crossbar networks attached to the same little board! Besides, for most applications the multidrop is more sensible—it takes far less hardware.

New Micros developed its family of HC11 boards (Forth in ROM) with a clever option—RS-232 or RS-485 console ports. The latter, with software-controlled line driver, is optimized for building a multidrop network in which the Hub always transmits to all nodes on one pair, with only one node at a time allowed to transmit on the other. There are various ways to go about this—New Micros offers a

clever piece of code with their boards called "Easy-A," which uses Control-A as a switch character to get all the nodes' attention, then follows that with a device address—the selected node then turns on its driver and the others continue whatever they were doing.

This and the corresponding tasker were in early development when we started the Microship project, but we were fortunate to engage the services of Forth wizard Bill Muench (author of eForth), who wrote the "Beeline" multidrop protocol code we use today. This uses an obscure feature of the HC11 to interrupt all nodes when the high bit is set, whereupon they all check the rest of the character to see who's the lucky one. (This "address-mark

wake-up," normally uses a ninth bit, but we have it set to the eighth.) Though this uses nonprinting characters and is thus less convenient during development from an ASCII terminal than Easy-A, it's tight and simple—we just set the high bit of the Node ID character ("T" for turret, and so on), squirt it down the line, and the link is made. A status LED on the selected node lights up for visual confirmation of network activity. Breaking all links, of course, is done by sending a nonsense character with bit 7 high.

In operation, the Hub (see Figure 2(b)), spends much of its time doing a node-scan, which updates an internal variable table every 15 seconds or so by sequentially selecting and querying all the nodes. In addition to collecting all variables that are needed by front-end displays and status monitoring, it also checks a watchdog character for every node—allowing a display on a small local LCD of the whole network's status as well as automatic restart of a crashed board.

None of this would work well without a multitasker, which allows the processors to handle all this ongoing communication without complicating the applications themselves. Nodes are collecting data, calculating statistics, controlling hardware, keeping an eye on internal sensors, and so on, all while the Hub makes the rounds, pretending to be a console and issuing simple Forth commands that request variables or execute commands relayed from one of the graphic front-end systems or the database engine in the server.

The nodes are built and coded as stand-alone projects, then dropped on the network as needed. The thruster controller, for example, power-cycles Hall-effect current sensors every few seconds to take a snapshot of all system loads and the instantaneous power available from the 480-watt solar array. This is folded into a running average that determines the available thruster power budget implemented as a "smart potentiometer" hacked into the Minn-Kota PWM controller. The throttle is then assigned that operating range—with a pushbutton override if emergency conditions (looming freighter, getting blown onto the rocks, whatever) justify pulling thruster power from the single Group 27 deep-cycle battery. (By the way, our available 32 amps should, under perfect solar conditions, yield 42 pounds of thrust for a cruising speed of roughly five knots without the aid of pedal or sail.)

Other nodes take care of video turret positioning and camera control, data collection, video crossbar switching, security, and so on—all relatively straightforward applications that operate autonomously with respect to the rest of the network, interacting only at the level of console

HASP[®]



Protects Your Software

Your software is your baby – and you want to look after it. You created it, you developed it, you saw it right through to the moment it was ready for market. Now protect it. 50% of business software is stolen; \$11 billion of developers' income is lost to piracy.* Is your software a statistic?

All over the world, more developers are protecting against piracy. They're protecting more products, on more platforms, with more security – and selling more as a result. And more of these developers are protecting with HASP.

To see why 25,000 developers worldwide protect with Aladdin, call and order your HASP Developer's Kit now!

HASP – The Professional Software Protection System*

- Rated #1 in industry tests
- Unequaled security
- Unparalleled flexibility
- Genuine ease-of-use and transparency
- Supported under Windows 95, 98 and NT, Mac, OS/2, NEC, UNIX, and LANs
- ISO 9002 quality and reliability
- New – USBHasp!





HASP Protects More

See us at Comdex Spring - April 20-23, Stand 1280

North America	800 223-4277, 212 564-5678, Email: hasp.sales@us.aks.com
Int'l Office	+972 3 636-2222, Email: hasp.sales@aks.com
Germany	+49 89 89 42 21-0, Email: info@aladdin.de
UK	+44 1753 622266, Email: sales@aldn.co.uk
Japan	+81 426 60-7191, Email: sales@aladdin.co.jp

Call us for a distributor near you!

* 1997 BSASPA Study
© Aladdin Knowledge Systems Ltd. 1997. HASP and USBHASP are registered trademarks of Aladdin Knowledge Systems Ltd. All other product names are trademarks of their respective owners.

1-800-223-4277

www.aks.com

ALADDIN

The Professional's Choice

AD LINK 421

(continued from page 58)

commands and condition-reporting variables. The resulting network is simple and robust, with minimal single-point failure potential.

Packet: Extending the Network to Wireless

Back when this project began, the network was straightforward: one boat, one multidrop chain of nodes, one controller Hub, and a graphic console implemented in HyperCard. As the mission profile evolved, however, we sold the yacht and began construction of twin micro-trimarans, autonomous yet linked human-scale vessels. This immediately introduced a nasty problem, not to mention the issue of scalability.

Each boat needs local control and monitoring, yet from the GUI and remote front-end perspective I want to see it all in one homogenous context. We need to wirelessly link the boats, but not at the LAN level—that would reintroduce a power problem by requiring 100 percent duty cycle of the Ethernet hardware and computers smart enough to use it. Speed in the control network is not critical, since time-sensitive tasks have their own processors.

The solution, known as "Packnet," not only takes care of boat-to-boat networking, but encompasses our backpacks as well: A mixed-mode packet and voice network on the 70-cm ham band. The Terminal Node Controllers (TNCs) run at a blistering 1200 baud, but draw so little

power that they can be left on permanently, as can the nodes that own them and the attached radios (hacked handheld transceivers). This network does not carry anything that can be classed as business traffic. The applications level that may include manuscripts and e-mail is entirely under FCC Part 15.

Node X is a Forth 68HC11 board like all the rest, but has a simple job that involves chatting with the other boat and both backpacks, shuttling Forth commands to the former, and collecting telemetry and GPS coordinates from all three. Normally linked to the TNC, it can also accept and verify a remote console login, then tell the Hub to route the authorized user via serial crossbar to take over the Hub's own console (or any other serial channel, including a port on the server).

Under normal conditions, Node X asynchronously (relative to the Hub's node-scan) passes queued commands to Node Y, while also collecting status and watchdog variables. Node Y, on Lisa's boat, is analogous to the Hub on mine—it runs the local show, with connections to other local nodes such as her thruster controller and power management.

One of the most amusing applications of all this is the tracking system, which allows either boat or either pack to determine range and bearing to any other. Between the boats, it's simple: Lisa's GPS hands continuous latitude-longitude data to Node Y via a small local serial crossbar, and the NMEA sentence is parsed into convenient variables passed to Node X (hence, the Hub) on every scan. If I want to know where she is, a simple inquiry invokes great circle calculations and displays or speaks the result (or shows her absolute location overlaid on the navigation chart). But the packs need similar functionality with a lot less available hardware.

The solution is a board from TAPR (Tucson Amateur Packet Radio Association, <http://www.tapr.org/>) called "MIC-E." Placed between a microphone and a radio, and connected to the serial output of a GPS, this device tails all voice transmissions with a 32-byte, 300-ms burst carrying latitude-longitude and basic telemetry/flag data—and it can be set to transmit at intervals if there's no voice traffic.

This completes the picture: While wandering around town, Lisa can find me or her boat by simply pushing a single button, thereby triggering a status request transmission that tells the Hub to compute range and bearing from the most recent GPS fixes from her pack and the other units. The speech synthesizer then powers up, links to the same radio via audio crossbar, and speaks the data she needs: "Your boat is 1.2 miles away on a bearing of 124 degrees..."

The longest continuously advertised C product in the history of ... mankind.

PC-lint

for C/C++
Version 7.5
presents Bug # 571

```
1 #include <iostream.h>
2
3 void f( char ch )
4 {
5     if( (unsigned) ch == 0xFF )
6         cout << "equal";
7     else
8         cout << "unequal";
9 }
10
11 int main()
12 {
13     f( '\xFF' );
14     return 0;
15 }
```

When this program is run, the usual output is "unequal", in spite of the attempt to cast away the sign extension. What went wrong? Call if you need a hint or visit our web site at www.gimpel.com.

PC-lint for C/C++ will catch this and many other bugs. It will analyze a mixed suite of C and C++ modules to uncover bugs, glitches, quirks and inconsistencies.

Version 7.0 introduced several spectacular and revolutionary innovations in the art of program validation. Taking clues from initializers, assignments, and conditionals, variable and member values are tracked, enabling reports on potential uses of null pointers and out-of-bounds subscripts. Almost 100 standard functions are rigorously checked.

Version 7.5 pushes the envelope of lint detection still further. User-defined semantic checking of functions offers a unique language to express argument constraints, inter-argument relationships, and return values. There are 75 new messages including cradle-to-grave checking of pointers and the latest admonitions and perorations of Scott Meyers.

Plus Our Traditional C/C++ Warnings:

Uninitialized variables, inherited non-virtual destructors, strong type mismatches, ill-formed macros, inadvertent name-hiding, suspicious expressions, etc., etc.

Full C++ Support - PC-lint for C/C++ is based on the ARM and is tracking the latest ANSI/ISO draft. It supports both Borland and Microsoft C/C++.

PC-lint for C/C++ \$239

Numerous compilers/ libraries supported. Runs on Windows 95, Windows NT, MS-DOS (built-in DOS extender), and OS/2. 30 day money back guarantee.

FlexeLint for C/C++

The same great product for other operating systems. Runs on all Unix systems, VMS, mainframes, etc. Distributed in shrouded C source form. Call for pricing.

Gimpel Software

3207 Hogarth Lane, Collegeville, PA 19426

CALL TODAY (610) 584-4261 Or FAX (610) 584-4266

www.gimpel.com

PA add 6% sales tax.

PC-lint and FlexeLint are trademarks of Gimpel Software

(continued from page 60)

Console Browsers and the High-Level Network

Once all these systems are chatting away in ASCII, we really don't want to look at them. Two years from now, floating down the Tombigbee River, I don't want to have to recall the format of the turret's SCAN and ZOOM commands to make the camera sweep slowly across the magnolia-draped waterfront of an antebellum plantation, then zoom slowly onto a sagging dock. But how do you write a GUI on an eight-bitter with 64K of address space and no disk?

Obviously, you don't. Let the little guys do what they do well, and add layers as needed to do the sexy stuff, which, after all, is consistent with our power-miser philosophy (when we're asleep, we need vigilant nodes, not animated GIFs).

The first implementation was easy: HyperCard on the console Macintosh. A crossbar front end that took two students two quarters to do badly in C (and a fat listing it was) took me an afternoon in Hy-

perTalk—and it was prettier and much more modifiable. However, I don't want to require that the Mac be on, or constantly switch between running applications to uncover the window that controls the boat. What we needed was a permanent, low-power graphic front end—preferably one that's portable enough to use via wireless from a pack on shore.

The second implementation was stunning in its beauty and integration—and written in NewtonScript. With Digital Ocean's Tarpon wireless Newton (no longer available) and a wired unit for the helm, along with software wizardry from Chris Burmester and Allpen's Erik Browne, we quickly evolved a clean custom layered control console that allowed pen control (via pen-base-T?) over the entire system; see Figure 3. In the turret view shown, touching any point on the circle around the boat moves the selected camera to that angle. (The NewtonScript code for the turret view is available electronically; see "Resource Center," page 3.) Drawing an arc

shades the selected region and scans the camera at whatever speed is selected by the slider. Other views control VCR or ham radio, display bargraphs of environmental sensors, and so on—one even lets us create macros (or presets) of Forth commands, name them, and select from a pick list. All tools simply send ASCII strings in response to touches, or convert received variables into graphs and icons.

This was nearly an optimal solution to the user interface of our control network, though it does involve one slight problem—custom code. Although the Newton is robust and power efficient, and although NewtonScript has the distinction of being the most elegant language I've ever used, the current uncertain fate of the platform (alas, Apple has announced it will no longer manufacture the PDA) may imply a shortage of developers and a future scarcity of upgrade paths. So how do we keep this same level of generality while achieving complete platform independence?

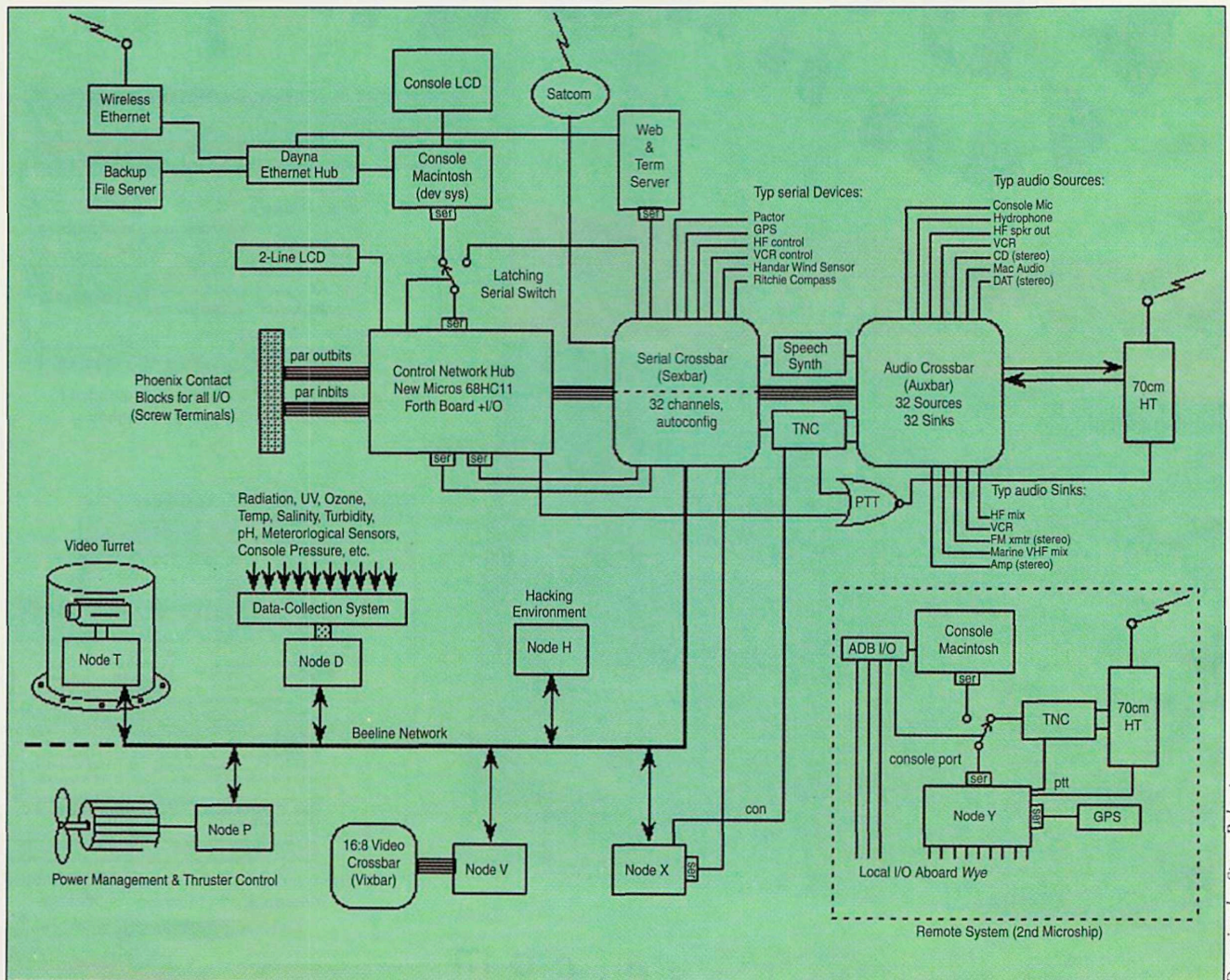


Figure 1: Microship control network brings numerous embedded controllers and random audiovisual and sensing devices into a single wireless graphic front end.

(continued from page 62)

The solution was under our noses all along! We are now building a web server and command engine that hangs on the serial crossbar so we can deploy browsers on whatever platforms are handy. At the moment, we are evaluating 3Com/US Robotics PalmPilot (<http://www.palmpilot.3com.com/>) running PPP via wireless modems to the boat—Symbol Technologies (<http://www.symbol.com/>) is introducing a waterproof version for industrial data-collection applications, and there are already a couple of browsers out there...and Java is due soon. Not only

does this let us have instant Telnet remotes while building the entire front end in HTML and Java, but it also opens up the application to a near-infinite range of volunteer developers, lets us upgrade hardware as product evolution turns today's magic into tomorrow's garage sale item, and lets us present all system documentation and online databases on the same handheld or console-mounted screens.

We do have a packaging problem to solve. A local control PalmPilot mounted in the console of a canoe-scale trimaran is going to get wet, and the normal sealing techniques don't work well with a pen

interface (maybe someone can write a mouse driver for us), but with our new-found platform independence, we could use another product in that role. But the architecture is simple: A low-power PC/104 embedded 486 PC running Linux, terminal server, and Apache sits atop the Hub, maintains the complete system state, serves code reloads when watchdogs detect failed nodes, and handles connections from the various wired and wireless browsers in our little nomadic family as we engage in pier-to-pier networking and knock back tankards of shoal draft after a hard day's solar pedalsailing.

And that, after all, is what it's all about—the bottom line is fun.

For More Information

If you'd like to receive more information on the Microship project, check our web site at <http://www.microship.com/>. Alternatively, you can register for a listserv in which we post a tech/adventure update via e-mail every month or two. Drop me a line at wordy@qualcomm.com and ask to be on the nomadness list.

Also, complete design packages on the crossbar networks (NRL-504) and video turret (NRL-505) are available for a nominal fee. These include schematics, listings, mechanical drawings, and lots of explanation. Registered owners receive e-mail updates as needed. Real soon now, I'll have similar monographs on the multidrop Forth network and tasker, web front end, Packnet, and power/thruster control. If you're interested in any of those, let me know and I'll drop you a line when they're ready.

DDJ

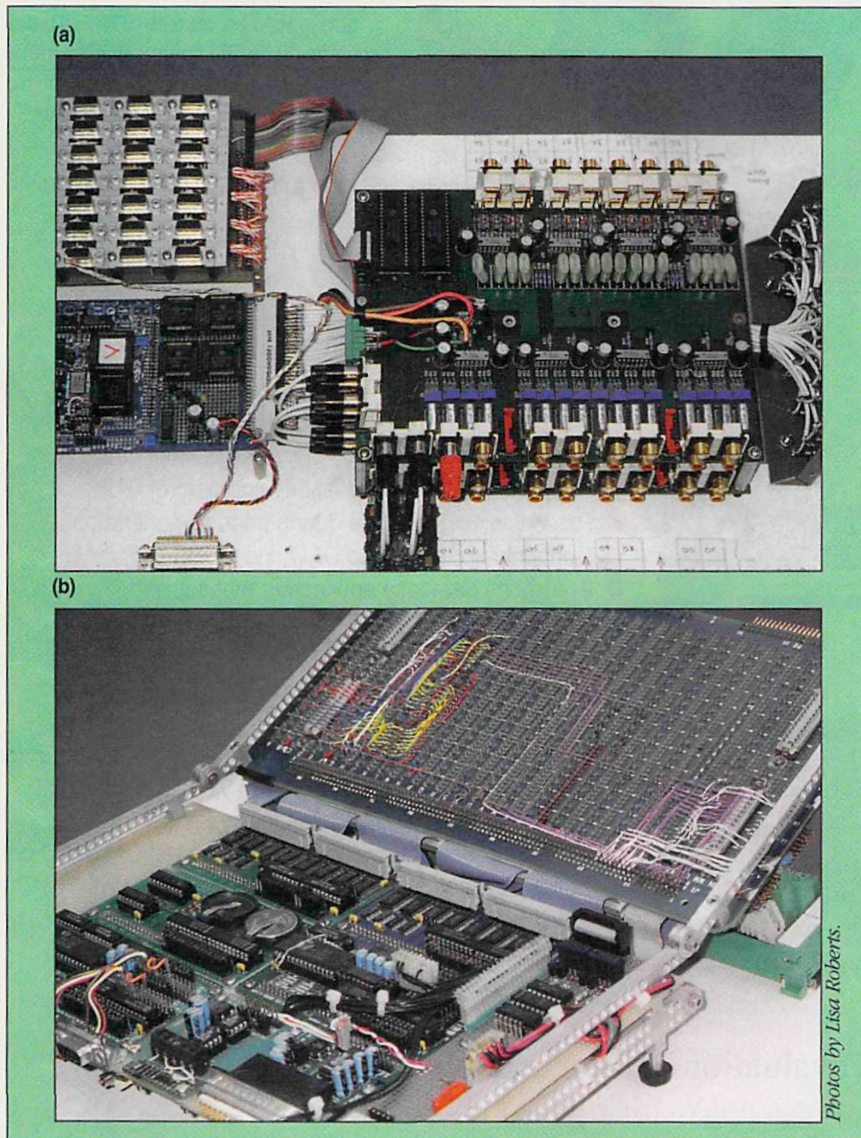


Figure 2: (a) Grand Central Station, a homemade audio, video, and serial crossbar network that is central to Microship resource management. DB-9 matrix of serial connectors is at upper left, built atop the Sexbar hardware; Vixbar is at lower left cabled to patch panel on far right; Auxbar is the large two-board stack. (b) The Hub, a folding assembly with the prototyping area (nexus) opened for service. The exposed layer is all New Micros 2x4 I/O boards bussed together with 34-pin ribbon; the Hub processor itself is underneath this panel. Peeking out from under the nexus is one of four Phoenix Contact distribution blocks that break out 34-pin ribbon cable to screw terminals.

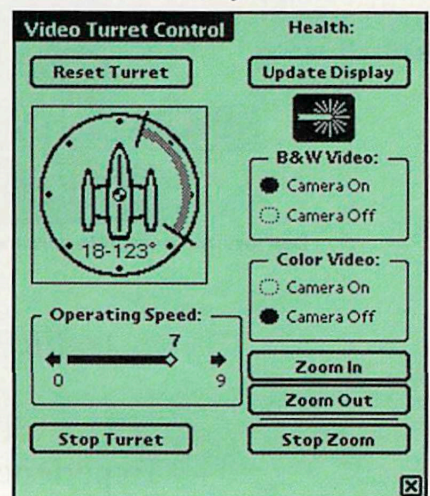


Figure 3: Newton turret-control tool written and designed by Chris Burmester. This is from a Digital Ocean "Tarpon," a wireless Newton that can control the Microship from anywhere within a few hundred feet.