WHY ASTRAL HAS THE HOTTEST TICKET IN TOWN

CANADIAN BUSINESS

Homegrown handguns

Toxins turn green

Air Canada's gamble

Harnessing the wind

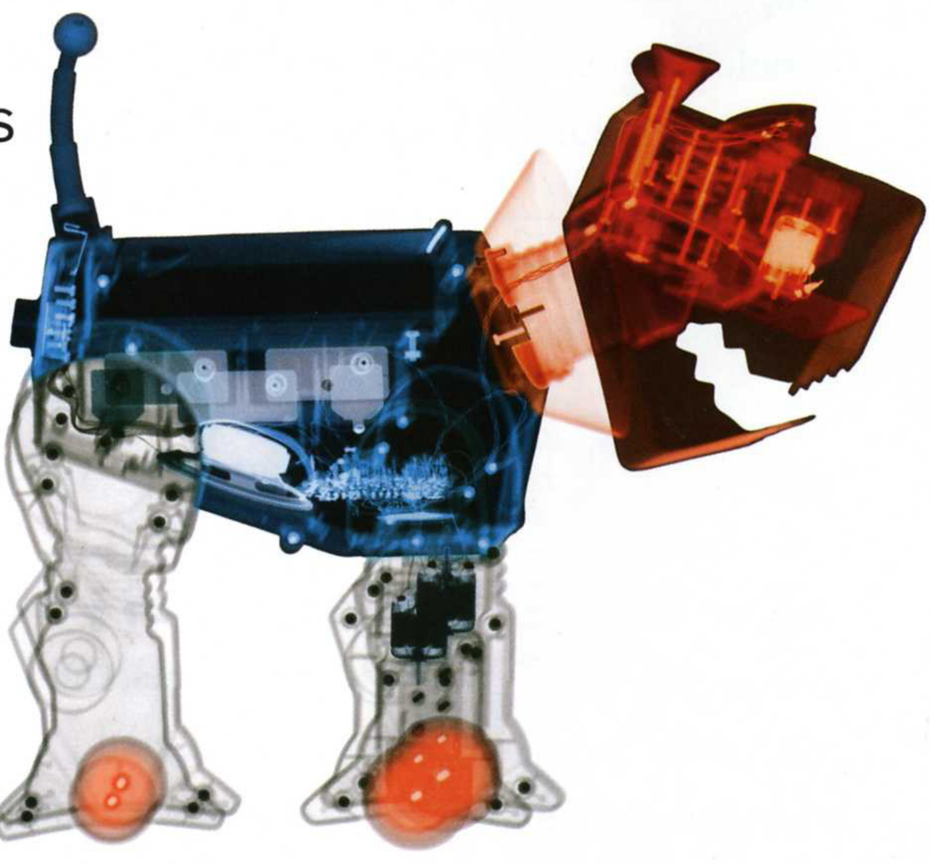
WWW.CANADIANBUSINESS.COM

SEPTEMBER 17, 2001

Cashing in on a new breed

Computer chips now run everything from cars to elevators to children's toys. In the not-so-distant future, they'll all be talking to each other

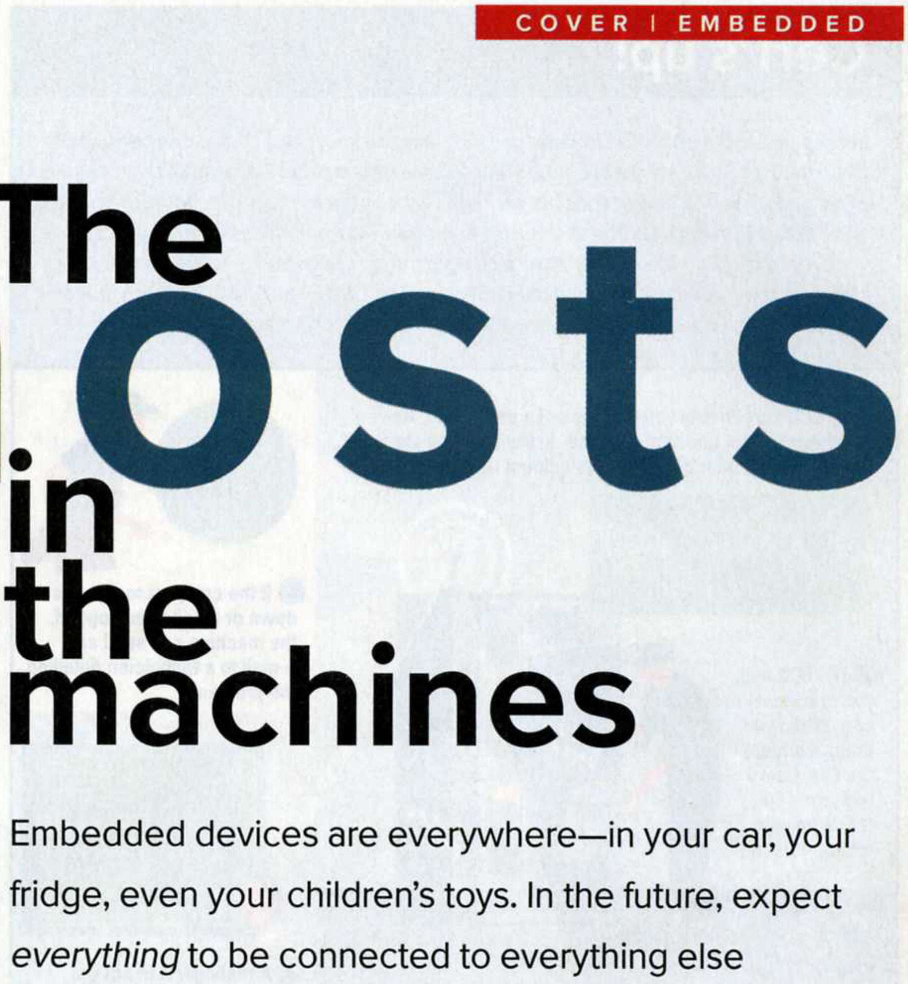
By Paul Kaihla



$4.25

38


Easy
PAY
GASOLINE
Snacks ·
0.00.00.00
step 1
step 2
RECEIPT
SLIDE CARD
4
97
C
O
L
A



The ghosts in the machines

Embedded devices are everywhere—in your car, your fridge, even your children's toys. In the future, expect *everything* to be connected to everything else

BY PAUL KAIHLA

Porter Burnette's look—a stained workshirt, concrete-covered construction boots—certainly doesn't scream "cutting-edge." At least at a distance, neither does the BP Connect gas station where he's filling up his black Chevy pickup, in a section of Atlanta that still bears vestiges of its blighted past (starting with the sofa dumped in the middle of the road a few blocks down). But as unlikely as it sounds, this spot, and Burnette's transaction, represent nothing less than a crucible of the future, a battleground in a struggle that will ultimately shape the next epoch of the Information Age.

Back in the 1970s, at stations like the Amoco that was just down the street, filling your tank was strictly an analog chore. A primitive meter in the fuel line was linked by gears to metal bands painted with numbers to rack up the cost of a fill. The personal computer was still a blip on the horizon; windows were what you rolled down to let fresh air into your car, not the product that created the fortune of the world's richest man. But through the '80s and '90s, even as the PC staked its claim on the future, something else was happening. In a fanatical drive to digitize, miniaturize and automate, engineers and developers had begun slipping silicon chips into the unlikeliest of objects, from dishwashers to elevators to talking dolls. Even gas pumps.

As Burnette squeezes the pump handle, chances are he's oblivious to the fact that a slew of microprocessors are helping to fill his tank. A calculator board on an Intel chip inside the pump reads electronic pulses from a meter in the fuel line, converts them to gallons, then transmits the fill in real time to a digital display on the kiosk's touchscreen monitor. A sensor in the kiosk's skimmer tells the unit's motherboard to read his Visa card and send the data to the chain's electronic transaction processing centre in Chicago for authorization.

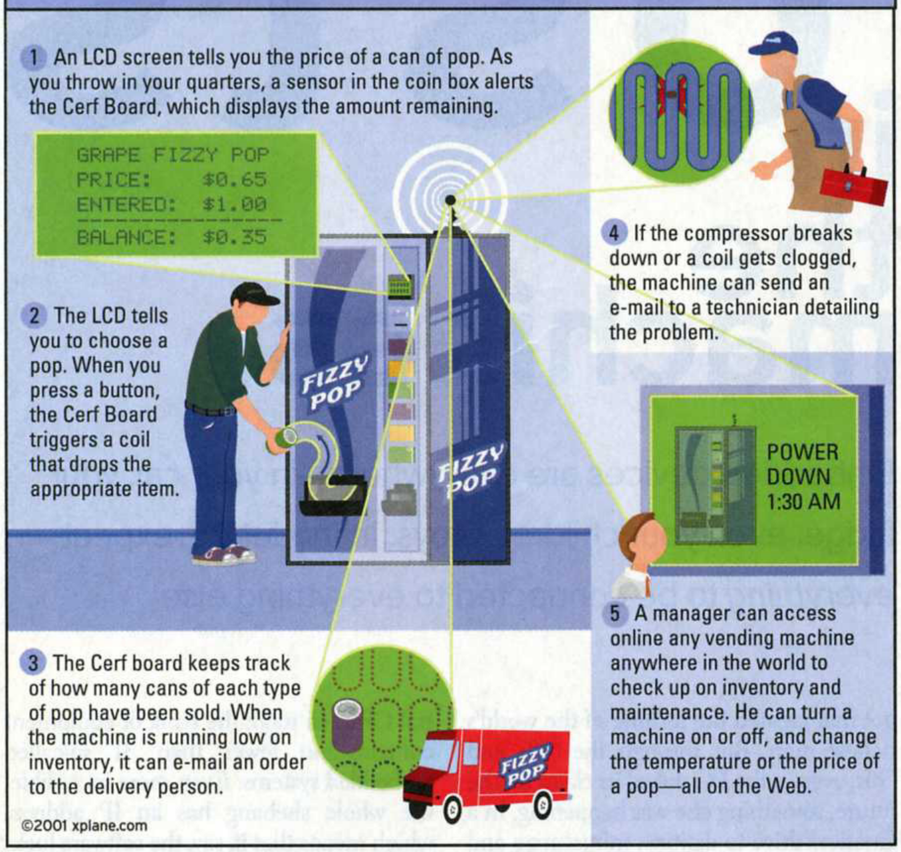
The pump in question is known as a Quantium fuel-dispensing kiosk, made by Tokheim Corp. of Fort Wayne, Ind., and it is ganged with nine other pumps on a central CPU. In total, the bank of equipment contains no fewer than 31 so-called embedded systems. Even more incredible, the whole shebang has an IP address, which means that if, say, the software locks up, BP managers can remotely restart the pumps from anywhere in the world. And that interactivity works two ways. The pumps' monitors are displaying Web pages from a server in Ohio, touting specials. "Pretty soon, when you dip your card, the system will know who you are," says Gray Taylor, Tokheim's vice-president of strategic programs and commerce systems. "The screen will say, 'Welcome, Paul, preferred customer of BP. Here are the stocks that you're tracking.' In fact, I'm talking to a company that wants to allow customers to insert a blank disc and burn a DVD movie right through our kiosks."

If mainframes led the first computer revolution and PCs the second, embedded systems represent computing's third Great Leap Forward. But the real revolution here may not be what you

JASON SCHNEIDER

Cerf's up!

Intrinsyc's Cerf Board can be found in products ranging from point-of-sale devices to medical equipment. It's essentially a souped-up 193-megahertz Intel StrongARM chip capable of giving 200 million instructions per second. That's about as much processing power as a Pentium 266 computer. The chip is also equipped with Intrinsyc's communication software, deviceCOM, which allows real-time data transfer between enterprise systems or between a whole bunch of embedded devices. The Cerf Board can run on any operating system, and it even has a Web browser. Here's how it works in a vending machine:



©2001 xplane.com

think it is. Despite the hoopla about all the wondrous things you'll be able to do with your PDA or smart phone, that kind of flashy consumer product constitutes only about one-tenth of the overall embedded-systems market. The future of "pervasive computing," as the phenomenon has been dubbed by everyone from IBM to Microsoft to Palm, actually lies in the many seemingly humble embedded devices that have already infiltrated our daily lives. Or, more precisely, in the networking of them.

What the PC did to bring the power of computing to the masses, what the cell phone did to liberate voice communications from ancient grids, the networking of embedded systems will do for data and machinery. This embedded cyberspace promises to turn your home into the technological equivalent of George Jetson's, and your business into Spacely Space Sprockets. Want to set the thermostat in your bedroom a few degrees higher before you leave the office? No problem. Need to recalibrate a robotic arm on the assembly line while you're in line for beer at a ballgame? Sure thing. Or better yet, the robotic arm can just go online and recalibrate itself.

So what, you may ask, is an embedded system anyway? Simply put, it's any object with silicon in it that's not a computer, server, mainframe or mammary implant. The key distinction is that a traditional computer is a multipurpose box. The job it performs is dictated by the software you choose to insert or download. An embedded system, on the other hand, has a hard-wired, dedicated purpose—locking and unlocking your car doors, for example—for the life of that unit.

Embedded systems constitute an almost invisible layer of computing that permeates the very fabric of daily life—so ubiquitous that they make the world as we know it go 'round. Motorola, the planet's No. 1 maker of embedded semiconductors, estimates that the average American encounters 150 embedded systems on any given day. The most obvious is the cell phone, but embedded systems also control your digital clock radio, your fridge's cooling and defrost cycles, and the elevators in your office building. The average car has 17 embedded systems, governing everything from the gas gauge and stereo to the antilock brakes. Every time you send an e-mail, that digital packet travels through dozens of embedded systems in the hubs, routers and switches that make up cyberspace.

These small yet omnipresent computers dwarf computers-in-a-box by sheer volume. Of the seven-billion-odd silicon chips that were fabricated, etched and implanted around the world last year, more than 90% ended up in embedded systems, not PCs. Sliced another way, more than 4.5 billion embedded devices were shipped in 2000 and tens of billions are currently in use; compare that to the installed base of desktop computers, which sits at about 510 million.

If the sheer scale of that universe blows your mind, buckle up. It's growing at a frantic pace, driven by the networking of embedded chips that are becoming smarter almost as fast as they're becoming cheaper. "The embedded market will become everything," says Richard Newton, dean of the College of Engineering at the University of California Berkeley. "It will subsume the desktop world. Embedded systems will ultimately displace the desktop for virtually everything except for very specific applications."

With engineers putting chips into more and more products—as well as boosting the number of chips in products that already contain them—the growth curve on the hardware side alone is mind-boggling. Most cars, for example, will soon have three times as many embedded processors as they do today. Significantly, processors in embedded devices are also getting smarter. In the past, most were low-cost, low-margin chips—generally four- and eight-bit versions, sometimes 16—which helps to explain why Intel has essentially been willing to cede the market to Motorola (a.k.a. the Intel of embedded chips). But that is changing. A survey conducted last year by *Electronic Engineering Times* revealed that nearly half of all designers of products that currently use 16-bit

XPLANE.COM

processors, and one-third of those whose products use eight-bit chips, are considering upgrading to 32-bit chips—the threshold at which an embedded device generally requires an operating system to juggle the several jobs it needs to perform. "The moment you say you're going to connect embedded devices, you're not going to slow down the Net to accommodate them," says Chris Rowen, CEO of the revolutionary Silicon Valley processor design firm Tensilica. "You're going to speed up the embedded machines, and they're all going to have to be 32-bit minimum."

That's why analysts are projecting that sales of embedded semiconductors will double over the next five years. It's also why the real Intel now wants a bigger piece of the action and has dropped US$5 billion in the past two years to acquire embedded networking companies. "For Intel," says group vice-president Tom Franz, "it has been the fastest growth area in terms of dollars spent."

make it tick; a few thousand lines of code sitting right on the chip do the trick. But an embedded device with a CPU that's 32-bit or higher almost always needs an operating system to juggle several jobs the machine needs to perform. That's the chip in the Tokheim pumps at the Atlanta gas station Burnette frequents, for example, and its operating system manages everything from the program that allows the kiosk skimmer to read credit cards to the software that pulls down Web content from BP Connect's server. "The sophistication of software required in pervasive embedded computing is one of the fundamental shifts that's taking place in technology right now," says Rowen.

But if all these embedded devices are going to talk, they need not only a standard Internet protocol to connect them, but also shared platforms for writing applications and leveraging the power of networking. And that raises a key distinction between

Of the seven billion chips that were etched and installed last year, 90% ended up in embedded systems—not PCs

While the proliferation of embedded silicon may indeed spell relief for slumping chipmakers, the real boom will be in software—for two reasons. First, it is often the major cost in any embedded system. The second reason is (you guessed it) the Web. Internet godfather Vint Cerf, a senior vice-president at Worldcom, predicts that by 2010, Web traffic generated by devices talking to each other could exceed that of people interacting over the Net. By 2050, he projects, there will be 110 billion machines on the Web—10 times the number of humans. This trend has already begun. In industrial automation, companies like Massachusetts-based Intellution are giving IP addresses to factory robots so managers can monitor or debug a plant floor over the Web. Carmakers are designing telematics that combine your dashboard CD player with a global positioning system. This year, Whirlpool is releasing the first Internet-connected white goods that can be controlled remotely by homeowners.

Given that these networked embedded devices need at least 32-bit chips, that means a hell of a lot of new demand for software, starting with operating systems. Consider this simple rule of thumb. The four-bit chip in a car's power door locks, say, does not need an operating system to

the traditional computer market—the mainframes and PCs that make up the industry's first two waves—and embedded systems. For the first 20 years, the vast majority of the operating systems (OSs) that run embedded devices were developed in-house by equipment manufacturers themselves. Motorola alone has built so many embedded OSs over the years—as many as 30—that its people have lost exact count. The ad-hoc fragmentation of embedded software platforms is coming back to plague the industry. Roughly half of the embedded world is currently running on thousands of proprietary OSs. The other half runs on software from no less than 50 commercial vendors. It is, in a word, Babel.

Sorting out the chaos—in other words, standardizing a platform for the next generation of embedded devices—has become an urgent mission at many Fortune 500 companies. Electronics engineering company Siemens, for instance, is gradually shelving its home-made operating systems as the market forces it to network its products. "Its customers are demanding remote access to things like its Saphir building controller, so they can update systems or diagnose problems remotely," says Rod Campbell, chief

CEO Jerry Fiddler suspects Wind River's operating system powers Raytheon's Tomahawk missile. "The government buys our software but won't tell us where it's going"

financial officer of Intrinsyc, an embedded solutions house in Vancouver. "And they're insisting that they not have to use proprietary tools to access these devices. Everybody is saying, 'I want you to use the common language that everybody else uses.' There are millions of people out there who understand HTML, but only a subset may know the Rockwell protocol or the Siemens protocol."

Buying off the shelf also shortens a product's development cycle by several months and usually constitutes an upgrade over in-house software. It's one of the reasons that Motorola, too, is increasingly going to commercial embedded vendors and has trimmed down its portfolio of in-house OSs to half a dozen. "If you're smart, you'll get rid of your proprietary in-house, home-brew operating system," says Gartner Dataquest's Tom Starnes, director and chief analyst of the global research firm's embedded microcomponents program. "If you bred it yourself, by now it probably looks like it's inbred."

The more that operating systems consolidate, the more that embedded applications will be leveraged. Or as Newton puts it, "The only time that any kind of computing has ever become pervasive is when there has been a common platform." Indeed, the Windows OS was just that in the desktop marketplace, which spawned a mushrooming array of software applications and in turn drove exponential PC sales. The same should hold true in the embedded universe: as applications proliferate, the software will become a far bigger market than that for operating systems alone. Over the next few years, the most bullish industry leaders expect the total market, which today stands at nearly US$1 billion, to grow to about US$10 billion.

Just who will own this market is another question. Of the 50-odd commercial vendors of operating systems in business today, most will be absorbed or left by the wayside. In fact, the real contenders have already been whittled to a handful. One of them, Wind River Systems of Alameda, Calif., has captured almost 40% of the market, largely due to the appeal of its core product, an operating system called VxWorks. Jerry Fiddler, Wind River's chairman and cofounder, launched the company in 1981 and now lives with the karmic implications of being an ex-Berkeley hippie who bids on defence department contracts. He says that VxWorks, which runs two-thirds of the world's 80 million laser printers, is probably the software in Raytheon's Tomahawk missile. But he can't be sure. "We're in a lot of black projects," he says, "which means that the government buys our software but won't tell us where it's going." Either way, Wind River expects to move from roughly US$400 million in sales today to US$6 billion within the decade—60% of that anticipated US$10-billion market.

Wind River's key competitive advantage lies in the fact that VxWorks has become the industry standard in hard, real-time operating systems (real-time being a measure of how quickly and consistently an OS can respond to inputs and prioritize functions). VxWorks's "hard" rating of one microsecond is a requirement for precision instruments like Tomahawk missiles. It's also essential in the hubs, routers and switches that comprise the Internet's infrastructure, which is why networking companies such as Cisco and Nortel make up more than half of Wind River's current business—and why "soft" operating systems like Linux and Windows simply aren't players in what has become the fastest-growing segment of the embedded market. "Is anybody going to put Windows CE

THOMAS BROENING

inside a network router?" Fiddler asks. "Nobody has, and nobody will."

Of course, networking is but one of the many industries in which the battle is being fought. And as software companies hustle to colonize the embedded universe, they are wrestling with the fact that each vertical market has its own requirements—and that the respective advantages of their operating systems vary accordingly. These requirements tend to divide along three axes: suit-friendliness, geek-friendliness and technical prowess.

In the first camp there is, inevitably, Microsoft. The company would dearly love to homogenize and hegemonize the embedded software market as it did the desktop, and is using the compatibility of its Windows CE platform with the rest of the Windows universe as a wedge into the market. Its open-source .Net initiative is, in part, an attempt to create a unifying interface for embedded devices, no matter what OS they have inside. (Embedded Java promises to do the same, which is why Wind River offers an assortment of Java development tools for its products.) "In industrial automation, you're going to see it very Windows-centric," says Derek Spratt, Intrinsyc's chairman and CEO. "There's no way a lot of the Fortune 500 companies are going to venture outside their comfort zone just for some added flexibility in their underlying operating system." Indeed, though most estimates of CE's market share still fall in the single-digit range, Microsoft's OS has enjoyed some wins. In the handheld PDA vertical, CE toppled Palm's monopoly by grabbing about a fifth of the market almost overnight—it is the OS in the pocket PCs released last year by Compaq, HP and Casio. The motherboard in the fuel dispenser at that Atlanta BP station also runs on CE, as does Intrinsyc's software for the Siemens Saphir controller.

But Microsoft's entry into the embedded space has been more opportunistic than organic: rather than building an OS specific to this market, the company has simply tried to insinuate its existing platform into it by rejigging successive versions of CE. And while Microsoft may engender a certain amount of knee-jerk loyalty among business types, it's important to remember that a lot of IT decisions aren't made by the go-slow suits, but by engineers—some of whom hold a deep antipathy to the Redmond, Wash., superpower. As those engineers are forced to move to a standard platform, many of them are turning to their favorite thorn in Microsoft's side: Linux.

This move is happening from two directions at once. First, both in-house engineers and independent developers are adopting Linux as an embedded OS. Meanwhile, a bevy of commercial vendors—notably Red Hat, MontaVista, Lineo and LynuxWorks—are selling embedded Linux design and support services. It's hard to track Linux's overall market share because distribution of the OS itself is free, but by some counts it is the fastest-growing operating system in embedded devices. The decision reached by Japanese elevator-maker Fujitec is typical. Last year, Sandor Markon, the company's manager of research, decided to port Fujitec's Supervisory Group Controller, a product that runs a bank of elevators, from the company's home-brewed OS to Linux. "We had to choose either an operating system by Microsoft or somebody commercial like that, or Linux," says Markon. "The reason we chose Linux is that it was free, we could have complete control, and there was nothing hidden. This product's life cycle is 20 years, so if we had to touch it after 10 or

15 years, we'd have the source code to do the modification. If we wanted to guarantee the same with a commercial system, it could become a nightmare. We could imagine the vendor in 2010, if still in business, staring at us with mouth agape when we ask them about bug fixes."

Nonetheless, with its market dominance and technical advantages, right now this game is Wind River's to lose. "There's no reason they can't pull it off," concurs Newton, the Berkeley engineering dean. "But whether it's a Wind River class of foundations that emerge or something else, it's important for something to happen in the embedded sector that is analogous to what happened with the desktop. We are absolutely going to see consolidation around dominant standards."

In other words, what's at stake in the battle for the embedded software market is much greater than simply who will dominate which market segment, or how long it will be before our gas pumps can talk to the onboard computers in our cars. What's at stake, to a large extent, is the future of computing and the Internet itself. The networking of embedded systems will not only dramatically increase the automation of the world around us, but also extend the Internet beyond PCs and PDAs into that automated sphere. Like a colorless, odorless force flowing through and between the countless artifacts of modern life, the embedded cyberspace will allow us to remotely, even wirelessly, access any facet of the engineered dimension of society, whether it's a technician receiving an e-mail on a cell phone that a gunked-up credit card skimmer in a Tokheim gas pump somewhere needs cleaning, or a networked vending machine signaling that it needs a refill. "Right now, when you're using that cell phone, you still know that you're accessing the Internet," says Wind River's Fiddler. "In the long term, you won't even be aware that you're accessing it because it'll be so pervasive that you're plugged in all sorts of ways at all sorts of times. It'll be as much a part of your life as electricity."

With that future in its sights, Microsoft is trying to position .Net as a service where handheld Internet appliances will automatically download enabling software when you, the subscriber, want to run an application like, say, your broker's trading program. But again, that handheld space is just one of dozens of verticals in the larger embedded market. They will change the shape of our world across the board with things like intelligent navigation systems, now being developed in Japan. Embedded systems in roadways will automatically steer cars in an orderly parade along a predefined course from, say, home to work and back. In the same way that those myriad embedded systems will continue to automate our lives, a much deeper change is underway: they themselves are becoming automated. In the industrial space, for example, developers are now designing machines that will look after their own health by combing the Internet for new protocols, software—even upgrades to the processor itself—and download them on the fly.

Ultimately, the embedded revolution will shift the nature of human communications and navigation from the obvious and explicit—think of dialing a cell phone to make a call, or typing an e-mail—to the invisible and implicit. But none of these new embedded technologies will make much headway in the current embedded world until it gravitates around common links and platforms. Only then will it become worthwhile for developers to write those futuristic applications for the masses. CB