

SILICON UPDATE

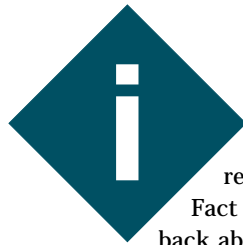
Tom Cantrell

Betting On Webware



Tom's rooting for the future of embedded

Internet apps, but he knows it won't be easy. There are some big obstacles to clear before reaching the goal. At least Lawrence Taylor isn't blocking the path!



have the utmost respect for Steve.

Fact is, he and I go back about 20 years and if it weren't for his drive and vision, I'd probably still be peddling chips with nothing more to look forward to than scratching my way up to VP of chip peddling.

Instead, I have a really fun job—poking around Silicon Valley looking for neat stuff to write about. Believe me, it's a lot nicer checking out new gadgets all the time rather than delivering the same old pitch day in and day out.

But, just because I admire Steve, doesn't mean we always see eye to eye when it comes to predicting the future. In fact, I just got off the phone after a bit of a flame session with him: does the embedded Internet hoopla fall more into the hype or hope category?

At last year's Embedded Internet Workshop, I got things rolling with a brief discussion of the major issues and challenges. I tend to think that when it comes to silicon, if it can be

done it will be done (and sometimes even should be done). Meanwhile, Steve is a show-me kind of guy.

Anyway, I firmly believe all manner of appliances will be on the web if the silicon wizards have their way. That's not to say it's going to be a trivial process to get from here to there. I see progress as more of a two-yards-off tackle than a long bomb.

But, the long drive starts with the first snap. Let's check our field position today, contemplate the playbook, and see what it will take to score.

PEEWEE PC

Although an Internet refrigerator may seem goofy, I could go to any computer store and put it together today with little muss and fuss. The fact is, the web is still overwhelmingly a home for PCs rather than appliances.

Thus, the brute-force approach to embedded Internet devolves to embedded PCs, a technology for which I have mixed feelings. It's a good news/bad news situation. The good news is that embedded PCs get to leverage the incredible software tools and know-how of desktop PCs. But, the bloatware tendencies of the software is the price to pay.

Just as quickly as hardware prices drop, software bloat makes up the difference. It's difficult to cobble together an entire EPC hardware and software solution for less than \$100—where it needs to be in order to boost appliance designs and volume.

Nevertheless, once you've got your pseudo-PC in place, software's a breeze. There is a huge variety of web-enabled RTOSs and such. Datalight offers a minimalist TCP/IP stack with direct support for low-end modems and serial lines. Don't forget that Linux (as Ingo's recent Real-Time PC columns showed) is quite a viable embedded option.

The hardware situation's a bit better with Windows CE, but you're still

going to need a rather hefty pile of silicon to get on the air. One unique advantage for CE is that it runs on other than 'x86 chips, notably the two most popular RISCs—MIPS and ARM.

The latter has been passing out licenses left and right and probably has close to a couple dozen major players signed up. Even Intel offers an ARM option (at 200 MHz, quite a speedy one) with their SA-1100, acquired in a deal to buy DEC's fab.

Annasoft has Windows CE running on SA-1100 and knows all the ins and outs of that port. Meanwhile, Mentor Graphics also fully supports ARM across the board. With the merger between the original silicon tool side of Mentor and the longtime embedded tool powerhouse Microtec, full support means silicon IP, development tools, and now CE adaption kits and more, all wrapped in a "seamless coverification environment."

How low can EPC go? One of the leanest platforms I've come across is the IPump reference design from Vadem, which you see in Photo 1.

IPump showcases Vadem's latest PC-on-a-chip, the VG330. This puppy is highly integrated with a CPU, real-time clock, 8254-compatible timer, dual 82569A interrupt controller, 16450-compatible UART with HP-infrared support, PCMCIA, and an LCD controller. It also has a built-in no-glue memory connection for SRAM, PSRAM, flash memory, DRAM, and SDRAM.

The '330 (based on the NEC V30) only runs up to 32 MHz, clear recognition that volume cost-sensitive apps are the focus rather than the desktop. Other practical concerns include power consumption, which the VG330 targets with extra power management (e.g., hibernate, doze, sleep) and clock control logic that cuts standby power to microamps.

Of course, the main criterion for boosting Internet-appliance volume is price. According to Vadem, a high-

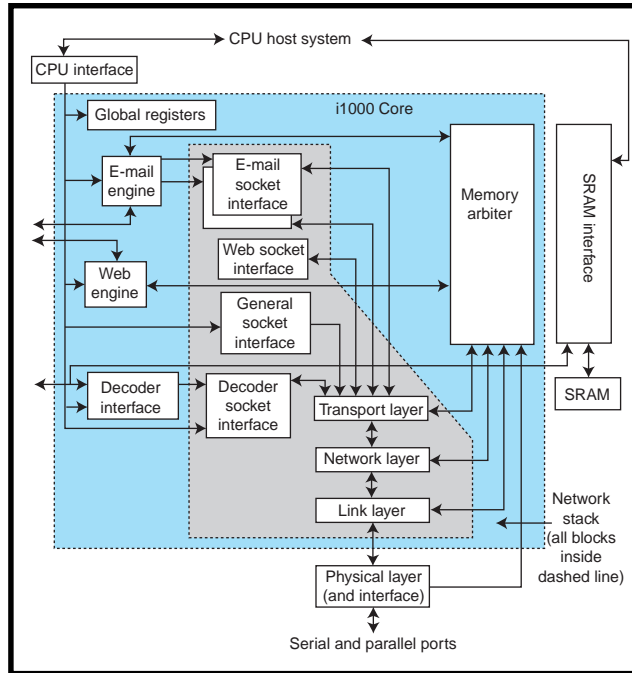


Figure 1—Delivered as IP and through licensees like Toshiba, the iReady i1000 implements protocol stacks in hardware.

volume OEM building a minimal variant (e.g., 14.4-kbps modem) of their IPump reference design could probably get unit cost down to \$50.

Part of the reason IPump is so inexpensive is because it relies on a DOS- (rather than Windows-) class OS and bypasses the eye-candy side of the web (i.e., http and html) in favor of moving data via simpler e-mail (smtp) and file transfer (ftp) protocols.

CUT THE FAT

There's just so far that you can shrink things if you require your Internet appliance to carry all that protocol baggage (SLIP, PPP, IP, TCP, http, ICMP, ftp, smtp, SNMP, etc.).

For goodness sake, all you want is to be able to check your fridge from the office to see if you should stop at the store on the way home. Seems like it shouldn't take more than an 8-bit MCU and a few screens of code.

As you've seen before, that's emWare's claim to fame. It's not a matter of Herculean code optimization. They just shove all the server bloat onto a gateway that handles the I-way on one side and lightweight point-to-point connections to the appliances.

GoAhead Software uses the same strategy. Each appliance requires only an 8-KB MicroAgent that hooks up

with the 'Net via the 400-KB gateway server. The server itself, known as Infusion, may be big on bytes but not on bucks. In fact, it's free for the taking (with source code) at www.goahead.com.

Both emWare and GoAhead make it easy to merge existing apps onto the I-way. Rather than a complete rewrite, all that's necessary is to specify which of the apps' existing data items should, in effect, publish and subscribe to the 'Net. Then, the agent code supplied by emWare and GoAhead is linked with your existing application to transparently (i.e., consuming only a small percentage of local processing power) move the data to and from the 'Net via the gateway.

These distributed configurations that shift the protocol processing and high-speed communication burden to a gateway may represent the best hope for truly consumer-class (and price) Internet appliances.

For instance, imagine the Internet refrigerator as just one node in the wired kitchen of the future. It and the other appliances could connect to a local gateway via dedicated links, perhaps power-line modem or short-range RF.

Does your refrigerator really need its own IP address? Probably not. After all, individual appliance data bandwidth requirements are low (you don't need streaming video to watch ice freeze) so these dedicated links can

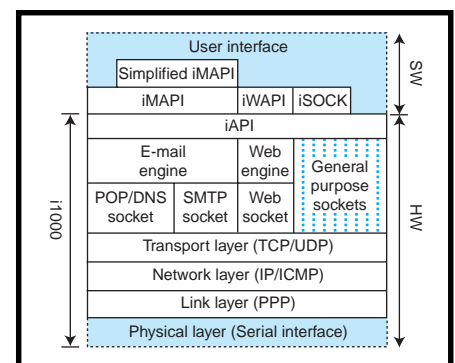


Figure 2—From the programmer's perspective, the i1000 APIs offer high-level mail, web, and socket services.

Listing 1—As this send-an-e-mail example shows, the i1000 makes it simple for practically any gadget to get on the 'Net.

```
#include "itypes.h"
#include "imapi.h"
#include "ihlp.h"
#include <stdio.h>

void main(void)
{
    ihlp1nit(0);
    im1nit(1000000L);
    printf("Talking to Email revision %x\n".imGetEmailRevision());
    imSetMailServer ("192.168.2.1");
    imInitiateConnection (IMF_SMTP);

    imSMTPSetFrom("\\"Test User\\"testuser@somewhere.com");
    imSMTPSetTo("\\"Another User\\"another@somewhere.com");
    imSMTPStartData();
    imSMTPSetData("From: Test User\r\n");
    imSMTPSetData("To: Another User\r\n");
    imSMTPSetData("Subject: Hello there\r\n");
    imSMTPSetData("\r\n");
    imSMTPSetData("How have you been lately?\r\n\r\n");
    imSMTPSetData("Sincerely,\r\n\r\n");
    imSMTPSetData("Test User\r\n");
    imSMTPEndData();
    imSMTPQuit();
}
```

trade off performance in favor of lowest possible cost.

IN THE CHIPS

Another trend in the race to “webify” everything is the emergence of highly integrated chips dedicated to the cause.

Ethernet may not be the right choice for the kitchen, but it’s a good match for office and (in the opinion of many) factory web gadgets. As for the latter, I explained in *Circuit Cellar 92* about how companies like HP opine that the ubiquity of Ethernet will steamroll nitpicking about whether it’s theoretically ideal for factory apps. It isn’t ideal, but it’s here, it’s cheap, and it can usually be coerced into getting the job done.

If Ethernet works for you, consider the Net+ARM from NETsilicon. As the name implies, it combines a 32-bit ARM7 CPU core with a 10/100 Ethernet MAC (media access controller). Included in the price (\$32.50 for 10k units) is a production license for the complete pSOS RTOS package from ISI. This package not only provides real-time hooks for your application but also includes the complete suite

of 'Net services including web (http), file (ftp), and mail (smtp).

Besides the CPU and MAC, the NET+ARM handles local processing with two UARTs, an IEEE 1284 (also known as a PC-type or Centronics) parallel port, and 24 parallel I/O lines. Of course, more app-specific I/O can be grafted on to the no-glue memory bus interface that hosts the external ROM and RAM, which contain the RTOS and networking software.

According to NETsilicon, about 80% of the ARM CPU bandwidth is available for local application processing. As well, they’re planning a number of variants, including low-cost versions that break the single-digit price barrier.

Before you rush off to call, be advised that the NET+ARM isn’t targeted to the experimenter or onesy-twosy buyers—unless you’re prepared to cough up \$20k for a development kit. Of course, the kit includes everything from a development board to a JTAG-based ICE and telephone support and training.

ARE YOU READY

The i1000 from iReady is another take on the web-chip angle. But, it’s arguably even less accessible to end users than the NET+ARM since the i1000 isn’t really a chip per se. Rather, it’s a bunch of intellectual property in the form of synthesizable logic written in Verilog. As shown in Figure 1, you can cut and paste the pieces you need to optimize gate count for a particular application.

iReady is investigating deals with licensees that might make standard merchant-market chips available at some point in the future. But for now, only ASIC designers need apply.

The i1000 virtual chip, unlike the NET+ARM, doesn’t include a CPU for application processing. It’s essentially a dedicated peripheral that can be added to any design, big or small.

The i1000 hooks into a system using various interfaces including a general-purpose bus that connects to your CPU and a dedicated RAM bus for web-related data structures (64–128 KB, depending on the number of



Photo 1—The Vadem IPump reference design uses their VG330 PC-on-a-chip to enable EPCs to get a piece of the Internet-appliance action.

protocols and sockets).

A generic physical transport interface lets you have it your way when it comes to exactly what kind of wire (or wireless) medium you prefer. Or, incoming data can be preprocessed and routed onto the chip via a separate decoder bus (e.g., JPEG, MP3).

At 20 MHz, the i1000 can move data at up to 10 Mbps and beyond. But for low-end apps, the clock rate can be cut significantly. The chip need only run at 35 kHz or so to keep up with a 28.8-kbps modem.

It won't be trivial to craft an i1000 ASIC, not to mention designing the rest of your appliance hardware. But, the hardware pain is more than made up with software gain.

That's because the i1000 does a lot more than offer a low-level socket-type interface. It handles much of the protocol-dependent processing that calls for lots of host-CPU headscratching.

For example, the i1000 automatically detects and processes many html tags on its own, reducing the burden on the host CPU. It does so by converting any recognized tags (unrecognized ones are passed up the ladder for your code to deal with) into an iReady command stream format (iCSF).

Instead of your application having to parse html tags like "<TITLE>" and "</TITLE>", the i1000 delivers a single-byte code saying in effect, "Here's a title," along with the text in between the tags. The i1000 does a lot of the grunt work associated with handling text, including attributes and fonts, via the iReady text encoding format (iTEF).

The application software interface in Figure 2 consists of APIs for e-mail, web, and socket services as well as lower-level register access. For example, using the iMAPI mail interface, it's a no-brainer for your app to send e-mail (see Listing 1).

SLIGHT MAKES RIGHT

Between peewee PCs, minimalist micro servers, and dedicated I-way chips, we're making pretty good progress toward a day when "Internet appliance" isn't an oxymoron.

Yeah, Steve, I know the two yards off tackle and a cloud of dust isn't as glorious as the long bomb, but I bet Internet appliances are going to put

points on the board sooner than you think. ☐

Tom Cantrell has been working on chip, board, and systems design and marketing in Silicon Valley for more than ten years. You may reach him by e-mail at tom.cantrell@circuitcellar.com, by telephone at (510) 657-0264, or by fax at (510) 657-5441.

SOURCES

TCP/IP stack

Datalight
(360) 435-8086
Fax: (360) 435-0253
www.datalight.com

IPump

Vadem
(408) 467-2100
Fax: (408) 467-2199
www.vadem.com

Net+ARM

NETsilicon
(781) 647-1234
Fax: (781) 893-1338
www.netarm.com

i1000

iReady
(408) 330-9450
Fax: (408) 330-9451
www.ireadyco.com

EMIT

emWare
(801) 256-3883
Fax: (801) 256-9267
www.emware.com

SA-1100 CE adaptation

Annasoft Systems
(619) 673-0870
Fax: (619) 673-1432
www.annasoft.com

SA-1100

Intel Corp.
(602) 554-8080
Fax: (602) 554-7436
www.intel.com

CPUs

ARM Ltd.
(408) 399-5199
Fax: (408) 399-8854
www.arm.com

MIPS Technologies, Inc.
(650) 567-5000
Fax: (650) 567-5150
www.mips.com

pSOS RTOS

Integrated Systems, Inc.
(408) 542-1500
Fax: (408) 542-1956
www.isi.com

Infusion

GoAhead Software
(425) 453-1900
Fax: (425) 637-1117
www.goahead.com

©Circuit Cellar INK, the Computer Applications Journal.
Reprinted by permission. For subscription information,
call (860) 875-2199 or subscribe @circuitcellar.com