 By Edward F Steinfeld, Consultant

TO CONVERT TEST-EQUIPMENT PRODUCTS INTO INTERNET APPLIANCES, YOU NEED THE RIGHT TOOLS AND SOFTWARE. YOU HAVE MANY OPTIONS; THE TRICK IS TO PICK THE BEST ONES.

Internet-appliance technology automates test equipment

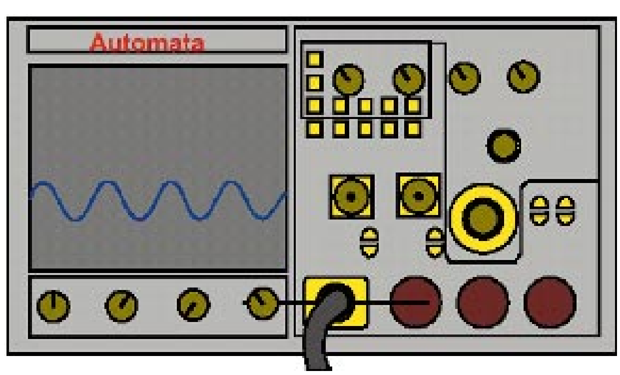
If you believe what you hear, everything from telephones to refrigerators to microwave ovens and even automobiles is becoming an Internet appliance. Internet appliances are instruments and devices that connect to computer networks to access, distribute, and store data. In many cases, you can manage and control the appliances from remote locations. Internet appliances use technologies that include the TCP/IP network protocol, embedded Web clients, and servers. In the near future, the list of underlying technologies will also include XML (eXtensible Markup Language) data framing.

If you put aside the hype, it is sometimes difficult to justify producing a networked appliance. Some appliances and instruments do not have the built-in computing power necessary to support network connections. The cost of the additional computing power and network interface can threaten to make a device noncompetitive. However, the extra cost is often justifiable. New capabilities allow access to new markets and can potentially offset higher costs. With a thoughtful design, a manufacturer can take a stand-alone tester product, add network support, and make five or more products from the same basic design. Moreover, depending on the product's complexity, if you eliminate the front panel, you can almost halve the manufacturing cost.

Most ATE (automatic-test-equipment) and laboratory-measurement instruments have been upgraded with enough internal computing power to support networked operation—usually with more power than they need to perform their basic functions. Turning such instruments into Internet appliances costs almost nothing and can even reduce the manufacturing cost. A device that already includes a network interface and a TCP/IP software stack is ready to undergo transformation into one or more Internet-appliance variations.

Why would you turn a successful product into an Internet appliance? One important reason is to permit users to share data with other computers in their companies. But a better reason is to enable remote control so that an operator can monitor and control the instrument from any office in the building. An embedded Web server makes such operation possible. The operator can use any Web-browser-equipped computer or terminal to communicate with the instrument. In effect, the user's PC and Web browser become the instrument's front panel. Other computers can still access the data. In fact, some Internet applications make sharing data easier and

Figure 1

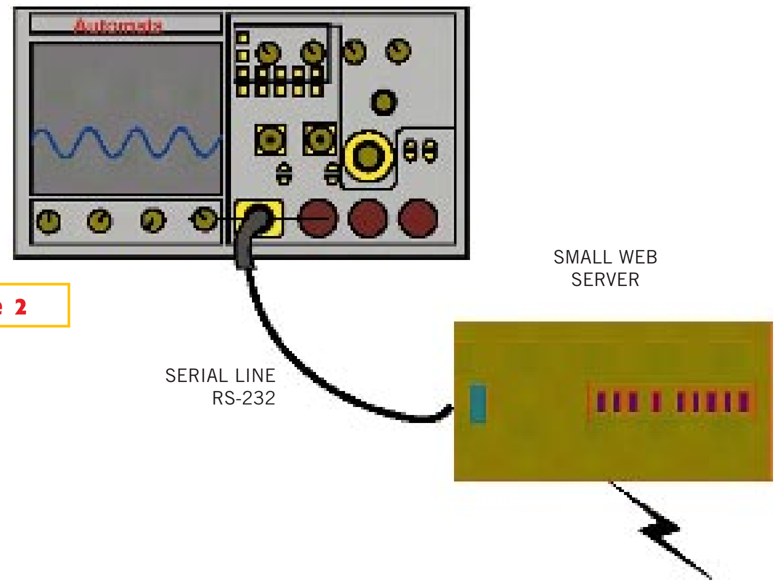


The block diagram shows the original instrument, the basis for a new Internet appliance. The manufacturer can sell an internally Web-capable product–with the networking turned off–as a replacement for this product.

more versatile than it is with more traditional methods of disseminating information.

Unless you want to create a proprietary network, you should choose Web technology as your Internet support. An Internet appliance that incorporates either an embedded Web browser or an embedded Web client can become a network client. This article explains some significant differences between a Web browser and a simple Web client.

Figure 2



A simple way to convert devices into Internet appliances is to use an external Web server. The instrument must have a port that makes setup and data-collection functions available.

More likely, though, the Internet appliance uses an embedded Web server that delivers HTML graphic pages to a Web browser on a user's PC somewhere on the network. The Internet appliance can become a Web server in several ways. Most end users already recognize Web technology's potential. Many instrument manufacturers, on the other hand, need to be convinced that the technology is sound and effective. An approach described later enables a significant manufacturing-cost reduction.

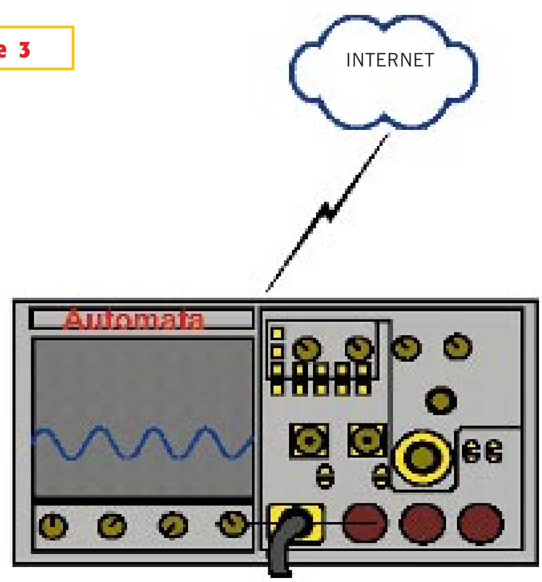
CHALLENGES OF NETWORKING

An ATE OEM faces several challenges when considering turning test equipment into Internet appliances. The first is the need to select a network operating system or protocol. Today, the choice is easy; if the device doesn't have to support an existing network, the choice is TCP/IP. These software stacks are readily available, and some are tuned for embedded and real-time operation.

Next, the manufacturer must make decisions about the computer systems and software with which the instrument must communicate. There are not only many kinds of computers, but also several operating systems. If users are going to interact with the instrument via networked computers, the product may have to support multiple user interfaces.

Even the connection to a file server can be troublesome. It is important to understand which file system or protocol the server uses—NTFS, NFS, FAT-32, FTP, or XML. Finally, distributing the user interface and keeping track of each newly connected user can present major challenges.

Figure 3



Some instruments need to communicate with information servers. Using an embedded Web client provides easy-to-use, two-way communication. This instrument can read and write objects with external Web servers and can use such options as XML and e-mail.

These networking challenges, which have confronted OEMs since the early days of distributed process-control systems, remain troublesome today. Solving networking problems is expensive when you use the older, traditional methods of development and interconnection. Rather than solving all of these problems, it is often easier to limit access to the device. In the past, OEMs developed their own proprietary network protocols and user interfaces to keep the total system manageable. Often, the system architects intentionally limited access to a reduced set of users and computers.

Today, there is an easy solution to nearly all of these networking problems. If remote users transfer data primarily for process and equipment monitoring and control, which usually means for equipment setup, the choices are Internet and Web technologies.

You can probably hear the customer's quality-control manager say, "I don't want my testers on the Web," or hear the production manager say, "With Web access, anyone can shut down my production lines." These comments result from a misconception. At least in industrial settings, Internet appliances use Web technology—*not* the World Wide Web.

Web technology is probably the most

widely used computer-network technology. Nearly every plant employee has used a Web browser either at work or at home. The user interface looks the same whether it is running on a Unix workstation, an iMac, or a PC. The Web-client software used in browsers can communicate with any server using the HTTP (Web)-server protocol. Even when they are generated on a variety of systems, the displayed HTML pages look alike. As with all technologies, you have choices. An Internet-appliance test instrument or system can be a Web client, a Web server, or both.

WEB CLIENT

The most common Web client is the ubiquitous Web browser—the one you use in your PC. The embedded Web browsers in industrial Internet appliances are smaller versions of the Internet Explorer and Communicator browsers. Embedded Web browsers are available from several vendors. One of the more popular embedded Web browsers comes from Spyglass (www.spyglass.com), the originator of the commercial browser. The purpose of the Web browser is to provide the user with a GUI (graphical user interface) to display information and collect user inputs.

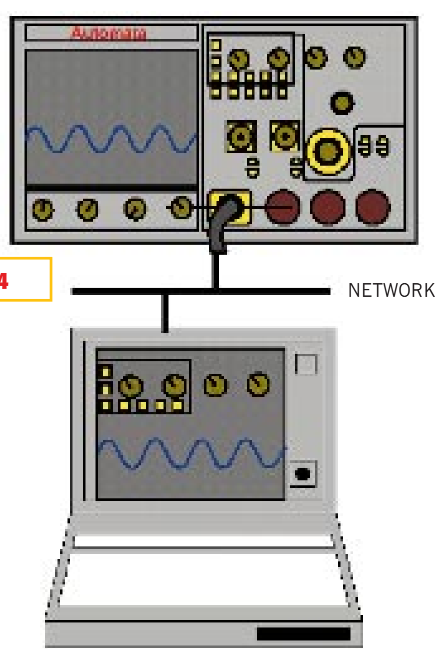
As a Web client, an appliance can transfer data to and from Web servers on the network. Web clients can transfer files using the FTP (file-transfer protocol) but are more likely to use the Web server protocol, HTTP. Even within FTP, you have choices. The latest in file formatting is the XML protocol. An Internet appliance can incorporate an embedded XML parser/framer. The XML framing protocol is a logical and human-readable way of identifying data fields and structures.

Most instruments that are being converted to Internet appliances already have human interfaces—front panels. Some of the more complex devices include embedded kernels with their own built-in HMIs (human-machine-interfaces). Some devices may not require an HMI or may use a Web browser to create the human interface. Few, if any, automatic testers that run as Internet appliances need to communicate with AOL, CNET, or Yahoo. These devices have limited and unique functions—to display specific data and to control testing. Thus, they have little need for a Web browser. Moreover, most embedded Web browsers require considerable memory and heavily load the processor.

The need for HMI compatibility between the local and the remote user panels might mandate the use of an embedded Web browser, however. The embedded Web client, without a browser, is an underused Internet application. Its memory requirements are modest, but it has much of the power of a Web server. The Web client communicates with Web servers by requesting objects from and sending objects to the embedded application. The memory requirement of an embedded Web client is about 12 kbytes, which represents a mixture of ROM and RAM.

Only a few companies currently offer embedded Web clients. Allegro Software Development (www.allegrosoft.com) has been shipping its WebClient software for a few months. The product is used in a telecommunications tester and a time-and-attendance terminal. Ebs Inc (www.etcbininc.com)has had an embedded Web client on its price list for a long time but does not indicate definite prices.

Figure 4



The most popular configuration of an Internet appliance adds an embedded Web server so that a remote PC running a Web browser can replicate the front panel.

WEB SERVER

The most common way to transform a device into an Internet appliance is to use an embedded Web server. This server delivers HTML-formatted graphics pages to Web browsers and communicates with other Web servers on the local network or the Internet. These graphics pages can display the data normally seen on the front panel of the instrument and can accept user input to control or set up the device. Most of the embedded Web servers, along with their HTML pages, fit in less than 100 kbytes of ROM and RAM.

The following paragraphs describe the creation of an Internet appliance using an embedded Web server. The costs and memory sizes listed are based on the Allegro RomPager products. Treck (www.treck.com), Prism (www.prism.com), Spyglass, and other companies offer competitive products. Many RTOS (real-time operating system) vendors offer embedded servers through alliances with these companies. Most embedded Web servers are provided as C source code, are processor- and operating-system-independent, and are licensed without runtime royalties.

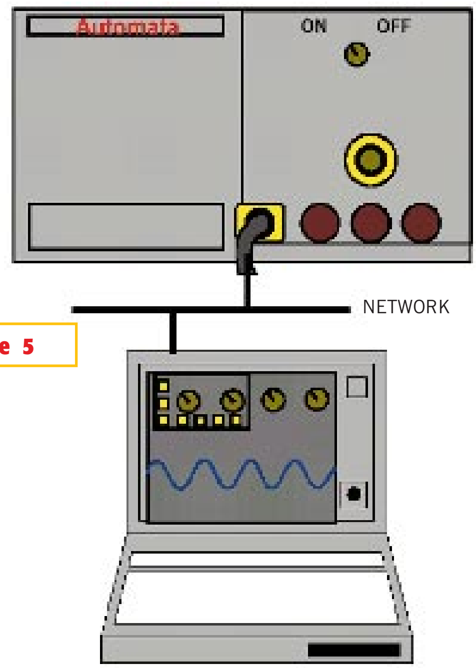
The key to using a Web browser to emulate a front panel is having the ability to display dynamic or changing data on the remote browser. Embedded Web servers provide this capability. Real-time data is provided to the HTML pages through buffers, variables, and functions. The embedded Web server connects this real-time or dynamic data to the requested HTML page and sends the page with the data to the remote Web browser. To alter the appearance of the dynamic data in the field, you must be able to change the pages on the fly. A runtime HTML page parser, such as Allegro's SoftPages, in the embedded Web server, enables such changes.

Most embedded Web servers work easily with embedded instrument-control applications. An embedded kernel or RTOS can be useful but is unnecessary for most embedded Web servers. Without an RTOS, just a few lines of code invoke the RomPager embedded-Web-server engine (**Listing 1**).

Most embedded-Internet tool kits are processor-independent and come with make files for the popular embedded kernels, operating systems, and TCP/IP software stacks. An Internet appliance can take the initiative in rectifying potentially critical situations if you add embedded e-mail. With embedded e-mail, a device that needs service can send a message to the person responsible for repairing that device or use e-mail to download software updates and services.

When using an Internet appliance that contains an embedded Web server, each of the end users needs a Web browser. Of course, most users already have browsers. Using Web technology provides the Internet appliance with a common network interface, a recognized user interface, and a connection to any platform or user. Developing Internet-appliance-based instruments is straightforward, and you need not be an Internet guru to do the job.

Figure 5



Removing the front panel from an Internet appliance is an excellent cost reduction. A user-supplied laptop PC running a Web browser now acts as the font panel. HTML pages and Java or ActiveX graphics replicate all the panel functions.

LOW DEVELOPMENT COST

Web-technology software costs less than you may think. The software tools and application code to develop a complete Web-enabled device with e-mail capabilities costs the device manufacturer no more than $50,000. Most embedded-Web-server software is licensed without runtime royalty fees. You can compare this $50,000 with the cost of implementing equivalent capabilities in house.

It should take two engineers approximately four months to have a reasonable embedded Web server running without support for all the RTOSs and TCP/IP stacks. The fully loaded cost for an engineer is about $150,000 per year, making the four-month, two-person effort cost about $100,000. Then, the engineers must still integrate the Web server into the product. With an Internet tool kit, this effort is much easier and faster than it would be with in-house-developed tools. Some tool-kit vendors have hundreds or thousands of embedded products running in devices. These commercial tool kits have been well-debugged and are maintained to current Internet standards. This level of reliability is difficult to duplicate in house.

One of the best parts of using Web-technology software is that there is no user software to distribute, no new software for different computers and operating systems, and no tracking of users. The HTML graphics and data displayed on the remote browsers are stored in the Internet appliance and sent, on demand, to the users via the network. The user provides the remote browser and computer. The HTML graphics can be displayed on any commercial browser.

The embedded Web server provides some security (usually log-on/password), and the corporation's firewall keeps outsiders away. Your customers are probably familiar with using a Web browser, so they need to learn only about the tester.

WEB-SERVER OPTIONS

Embedded Web servers come in versions from minimal to full-featured. The amount of memory a server requires depends on its features. Most projects will probably use fairly full-featured servers. The additional cost and hardware requirements of such servers are modest.

Most engineers don't usually consider e-mail an option for use in a machine controller. Web server vendors can elaborate on the possibilities and advantages of e-mail. A feature of embedded e-mail is the ability of the test equipment to send e-mail to alert a supervisor to potentially dangerous conditions. Another feature is the ability to update an instrument's internal software by sending updated code to the instrument as an e-mail attachment.

Some embedded Web servers support DNS (Domain Name Services), network time synchronization, and XML parsing and framing. The additional memory needed for each of these options varies from 5 to 20 kbytes of ROM plus 5 to 10 kbytes of RAM.

A runtime HTML parser permits the design of custom Internet-appliance

LISTING 1–ROMPAGER EMBEDDED-WEB-SERVER ENGINE

```
TheTaskDataPtr = AllegroTaskInit();
while (theTaskdataPtr != (void ) 0) {
      /*
            Wait here for time or event message.
            And give up time to other threads.
      */
      if (!AllegroMainTask(theTaskDataPtr, &theHttpTasks, &theTcpTasks)) {
            AllegroTaskDeInit(theTaskDataPtr);
            theTaskDataPtr == (void *) 0;
      }
}
```

front panels. Users can create displays that meet their needs and can customize the displays with distinctive logos. An equipment vendor must provide the customer with the variable names and formats that are used within the instrument. A runtime HTML parser can provide an instrument manufacturer with a new source of revenue—writing custom pages for customers.

An XML parser/framer can give the appliance a state-of-the-art method of sending and receiving data. It can parse XML-formatted data and can pass the information into C-programming constructs. Equipment manufacturers can use these capabilities. Web technology provides a menu of features from which you can select any combination with reasonable assurance that the choices can cooperate with one another.

Vendors of embedded Internet products include:

- Allegro Software Development, which provides advanced and basic embedded Web servers, an embedded Web client, embedded e-mail, an XML parser/framer, and an HTML runtime parser (products with similar parameters are available from other vendors);
- Ebs, which offers a minimal embedded Web server, a TCP/IP stack, and a new embedded browser;
- Phar Lap Software (www.pharlap.com), which has a proprietary Win32 RTOS, TCP/IP stack, and embedded Web server;
- Spyglass, which ships both an embedded Web server and an embedded Web browser; and
- Treck, which sells both a TCP/IP stack for embedded and real-time applications as well as an embedded Web server.

Go to Internet World's WebCompare Web site, webcompare.internet.com, for a partial list and comparison of available Web servers.

INTERNAL COMPUTING POWER

Most instruments and controllers that are Internet-appliance candidates are complex and based on a microcomputer chip. Whether that chip is an 8- or a 16-bit device is not important. If the microcomputer chip can support the network protocol, it can support the Web-technology software that can turn the instrument into an Internet appliance. Some vendors use the 8051 microcontroller, but an 80186 or one of the many 32-bit processor chips is more common. The TCP/IP software vendors can tell you whether their software runs on the microprocessor chip your instrument uses.

For instrument conversions, the instrument must use a powerful enough microcomputer and must contain the necessary network-interface hardware. The network hardware can be either a serial port or an Ethernet connection. A serial port is less desirable and requires the instrument to have an external modem or an external network server.

If you design an instrument from scratch, you might consider using microprocessors from companies such as Connect One or NetSilicon. These chip vendors include some of the network and Web protocols in their microprocessors.

An instrument equipped with a serial port can connect to an external Web server. From there, the instrument functions can be made available to other machines on the network. There are a few manufacturers of these small Web servers. Dawning Technologies (www.dawning.com), designed one of the earliest small, portable Web servers. With most external servers, the instrument manufacturer must write software that communicates with the instrument through the serial port and must design the pages that are used for controlling and monitoring the instrument.

The advantage of this approach is that the instrument manufacturer need not modify the product to make it into an Internet device. The disadvantages are the added cost of external hardware and limitations on the use of the serial port. The hardware cost difference is similar to the cost difference between internal and external data/fax modems for a PC.

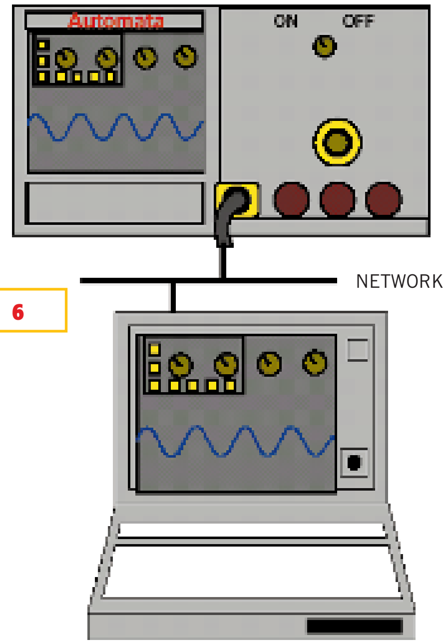
For the end users, an external server is a good approach if the instrument has no network connection but is controllable through the serial port. Functions are limited to those that you can access through the serial port. Most instruments designed in the past few years have a network connection, usually Ethernet and TCP/IP. These instruments can send data to external computers for further processing and storage. Usually, you cannot set up and control such instruments via the network, however.

With basic network functions built in, the instrument can become an Internet appliance simply through the addition of an embedded Web server. You may need additional ROM and RAM, but most instruments of recent design have a surplus of memory and can, if necessary, accommodate memory chips of greater capacity without a redesign.

WEB SERVER INSIDE

With a Web server inside the instrument, you can design HTML graphics pages that look like the original hardware. You might use sliders to replace rotary knobs and clickable pushbuttons to replace toggle switches. You can replicate CRT-like displays and analog meters if you use Java- or ActiveX-driven graphics. A Web browser can remotely display

Figure 6



Users often request that the front panel and the remote Web browser provide identical interfaces to the Internet appliance. An embedded Web browser driving a laptoplike front panel can achieve this feature, albeit at high cost.

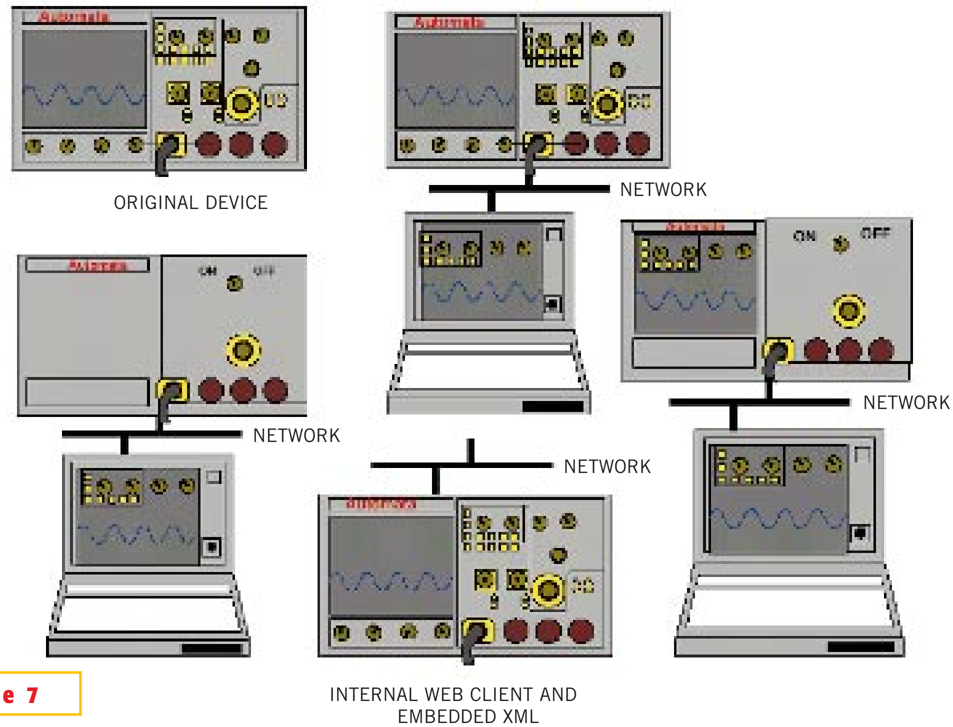


Figure 7

From a basic design, you can create five products that form a family of Internet appliances.

the entire human interface. You can create the pages for the Web browser by using such tools as FrontPage (www.microsoft.com) or Dreamweaver (www.macromedia.com) HTML-page generation software.

The instrument manufacturer must write code to communicate between the Web server and the instrument's various functions. This code consists of pointers to buffers, variables, and functions. To include the dynamic data in the HTML pages, vendors insert the variable and function names in the HTML code. Vendors accomplish this insertion in different ways. Most approaches separately identify the dynamic data variables and the HTML code.

When the Web server and HTML pages are complete, all of the front-panel functions, controls, and displays are available to the operator's browser wherever network access to the instrument is available. The network might be accessible on the bench alongside the test equipment or in an office two buildings or two countries away. A quality-control manager can monitor test results in real time while the parts undergo tests on the production line.

Once an instrument has an Internet connection and the users have access to all of the front-panel functions through their PCs, you can eliminate the front panel because they are expensive to build and must receive support from the instrument manufacturer. If HTML graphics pages on a Web browser can replicate the panel functions, the panel may need only an on/off switch.

A typical instrument panel contains a set of knobs, switches, and meters. The knobs and switches set up the device so that it can measure or monitor external signals and select the method of displaying the monitored values. A front panel full of controls and displays is expensive to manufacture and can contain an expensive display and electromechanical devices that are usually hand-wired. For many instruments, the cost of the front panel is half the cost of the entire instrument. In many cases, you can slim down the instrument's power supply if it no longer must power a hungry video display.

If you make an instrument into an Internet appliance, you can give it added capabilities at a fraction of the original cost. Customers can select the type of computer best suited to their needs and supply the front panel through their Web browsers. The user can choose the location of the front panel. Multiple users can monitor the instrument while one user retains control. The corporate firewall provides the primary protection against outside intruders, but the Web-server tool kit can supply additional security features.

CREATING PRODUCTS

You can turn one test instrument into several Internet appliances. The same basic chassis, electronics, and internal software can yield at least three and possibly five products. This versatility should make the Internet-appliance concept especially attractive to instrument manufacturers. To create the appliance, you should start with a stand-alone machine controller or instrument (**Figure 1**) and add the processing power, extra memory, and network hardware and software interface and choose any of several Internet applications.

You can keep the original product or upgrade it with a higher speed processor, a network interface, and additional memory. Then, you can include the Internet software and not activate it. This product now becomes the basis for four additional Internet appliances.

You can use the original unmodified instrument as an Internet appliance. If the serial port allows access to the setup functions and output data, you can connect the serial port to an external Web server, making the instrument accessible via a Web browser anywhere on the network. This option is usually used with older instruments. Users can also connect existing devices to the network in this way (**Figure 2**).

Now, you should add a network interface, a software stack, and an embedded Web client. With the embedded Web client, the appliance easily communicates with information servers to download setup commands and to upload data. You can add XML parser/framer software to make the information more accessible. The XML software works in any of the Internet-appliance configurations (**Figure 3**).

Instead of adding simple client software, you can add more robust embedded-Web-server software so that remote Web browsers can display data and control data transfer to and from the appliance. Users with desktop PCs, laptops, or any Web terminal that includes a full-featured browser can access the Internet appliance. HTML pages with dynamic data capability provided by the embedded Web server can replicate the front-panel functions. From a remote browser, a user can manage, set up, and display data from the Internet appliance. Java applets, usually provided by the embedded-Web-server vendor, can create a graphical display that imitates CRTs, analog meters, or chart paper (**Figure 4**).

To produce a cost-reduced version of the Internet appliance, you can remove the front panel and use only the remote browser to replicate the expensive panel. Often, the front panel is a tester's most expensive subsystem. The panel contains electromechanical components, an LCD or CRT, and many rotary and toggle switches. Software can replicate all of these parts on HTML pages. If you replace the front panel with a user-supplied PC and Web browser, you can halve the cost of manufacturing an Internet appliance (**Figure 5**).

Finally, a more expensive option is to use an embedded Web browser inside the Internet appliance and to use the same HTML pages on the front panel as on the remote browsers. You can replace the panel with an embedded PC, which is functionally equivalent to a laptop computer running a full-featured Web browser. The front panel's look and feel are identical on the local appliance and on a remote browser (**Figure 6**).

FIVE PRODUCTS FROM ONE

Using common components reduces the cost of the Internet appliance and simplifies manufacturing, stocking of parts, and service. All versions look similar and have the same user interface. With Web technology in your machine controller or instrument, you can produce five products, all with nearly identical electronics and software (**Figure 7**). You can design the basic unit with the appropriate memory and software support but without the external network connection. This product looks identical to the original instrument. If you add the external network connection and turn on the Web server software, you have an Internet device with many additional capabilities. Finally, if you remove the expensive components on the front panel and install a new front panel, the result is a lower cost product with the same capabilities as the original product.

You can also create a follow-on revenue stream by providing options such as e-mail. For example, some runtime HTML page parsers permit your customers to modify the look and feel of the system and enhance it for their own installations. You might even provide a service that generates additional revenue by adding customer logos and new functions. Turning automatic test equipment into an Internet device should increase the market size, sales, and margins at almost no cost.

Web technology makes sense for a host of reasons that are not just economic. Browsers are so commonplace that most users are familiar with them and can use them without training. Indeed, users find themselves drawn to Web-based products. Moreover, thanks to Web technology, all instrument software can now reside within the instrument, eliminating distribution problems and the possibility of illegal copying.□

AUTHOR'S BIOGRAPHY

Ed Steinfeld has more than 25 years' experience in real-time and embedded computing. He began his career as a programmer, writing code and designing hardware to test hybrid circuit boards for Picker X-Ray (Cleveland). He has marketed embedded and real-time products to OEMs and resellers for Digital Equipment Corp, (Maynard, MA)—now part of Compaq Computer, (Houston)—VenturCom Inc (Cambridge, MA), and Phar Lap Software (Cambridge, MA). His international experience includes a stint in Hong Kong as a Far East channels manager for international OEM sales in Europe and the Pacific Rim. He now provides market research, planning, and services to the embedded-computing industry.