

Add Internet Appliance Technology to Automatic Test Equipment

We hear that everything from telephones to refrigerators, microwave ovens, and even your automobile is being turned into Internet appliances. What is an Internet appliance and what technologies are used? This article tells you more about the Tools And Software Needed To Convert Existing Test Equipment Products Into Internet Appliances.

WHAT IS INTERNET APPLIANCE TECHNOLOGY?

We hear that everything from telephones to refrigerators, microwave ovens, and even your automobile is being turned into Internet appliances. What is an Internet appliance and what technologies are used? An Internet appliance is an appliance, instrument, or device that connects to a computer network to access and store data and may allow the management and control of the appliance remotely. The technologies used include TCP/IP network protocol, embedded Web client or servers, and in the very near future, XML data framing.

Aside from the hype, it is sometimes difficult to find the justification to produce a networked appliance. Many existing appliances and instruments do not have the built-in computing power necessary to support a network connection. The added computing power and network interface is a cost that Edward Steinfeld Page 2 of 19 Automata International Marketing 22-March-2000 could make the device noncompetitive. However, there are justifications to invest in this cost. In fact, much of the cost can be offset by new functionality and new markets where the product can compete. With thoughtful design, a manufacturer can take a standalone product tester, add the network support, and make five or more products from the same basic design. In one case, the manufacturing cost of the product can be nearly halved by the elimination of the front panel.

Most automatic test equipment (ATE) and laboratory measurement instruments have long been upgraded with higher internal computing power. Usually with more power than is needed to run the existing functions. Turning such an instrument into an Internet appliance is almost free and possibly less expensive to manufacture. If the device already has a network interface and TCP/IP software stack, there is little left to make it into one or more Internet appliance variations. Why would one turn a successful product into an Internet appliance? The most obvious answer is to share the data it collects with other computers in the user's company. But, a better reason is to remote the front panel allowing the operator to be at a desk or even in another building and still monitor and control the instrument. An embedded Web server makes this possible. The operator would use any computer or

Web terminal that has a Web browser to communicate with the instrument. In effect the users PC and Web browser becomes the front panel of the instrument. Oh, one can still share data with other computers and there are some Internet applications that make this data sharing easier and more versatile.

INTERNET APPLIANCE OPTIONS

Unless one wants to create a proprietary network solution the obvious choice of Internet support is the use of Web technology. An Internet appliance can become another client on the network by using either an embedded Web browser or embedded Web client in the appliance. There are major differences between a Web browser and a simple Web client; these will be described later.

The Internet appliance will more likely use an embedded Web server delivering HTML graphic pages to a Web browser on a user's PC somewhere on the network. There are a number of ways the potential Internet appliance can become a Web server. One of the possible solutions described later can have a significant manufacturing cost reduction. First, the manufacturer must be convinced that using Web technology is a sound and effective solution. Most of the end user's are aware of the potential of using Web technology

CHALLENGES OF NETWORKING

What challenges does an ATE OEM face when he is considering turning his test equipment into Internet appliances?

First, he needs to select a network operating system or protocol. Today the choice is easy. If the device has no baggage of an existing network to support, the choice is TCP/IP. TCP/IP software stacks are readily available and some are tuned for embedded and realtime operation.

Next, comes the problem of what are the computer and software platforms the instrument will have communication? Not only are there many different kinds of computers, each computer may be running a different operating system.

If the connected computers are to have human interaction with the instrument, the OEM may have the problem of supporting multiple user interfaces.

Even the connection to a file server can be a problem. What file system or protocol does it use? NTFS, NFS,

FAT-32, FTP, XML, or what?

Finally, a major challenge is software distribution of the user interface and keeping track of each newly connected user. These networking challenges have been with OEMs since the early days of the first distributed process control systems and they remain problems today.

Networking problems are expensive to solve when using the older traditional methods of development and interconnection. Often it is easier to limit the access to the device rather than solve all these problems. In the past, OEMs developed their own proprietary network protocol and user interface just to keep the total system manageable. Often they had to restrict access to a reduced set of users and platforms.

Today there is an easy solution to nearly all of these problems. If the total system of extended users is primarily for data transfer, monitoring, and control - usually setup, the solution is the use of Internet and Web technologies.

WEB TECHNOLOGY

Can you hear the 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."

The solution is Web Technology, NOT the World Wide Web.

Web technology is probably the most widely used computer network technology. Nearly every plant employee will have used a Web browser either at work or at home. The user interface looks the same whether it is running on a UNIX workstation, iMAC, or PC. The Web client software used in the browsers can communicate with any server using the HTTP (Web) server protocol. The HTML pages displayed can likewise be generated on a variety of systems and all look alike. As with all technologies, one has choices. The Internet appliance test equipment 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. Embedded Web browsers are used in industrial Internet appliances and are smaller versions of Internet Explorer or Netscape browsers. Embedded Web browsers are available from a number of vendors. One of the more popular embedded Web browsers comes from the originator of the commercial browser, Spyglass. The purpose of the Web browser is to provide the user with a graphical user interface (GUI) to display information and collect the user inputs.

As a Web client, the appliance can send to and receive data (objects) from Web servers located on the network. Web clients can transfer files using FTP protocol but more likely will use the Web server HTTP protocol. Even within the file transfer protocol, one has choices. The latest in file formatting is the Extensible Markup Language (XML) protocol. The Internet appliance can have an embedded XML parser/framer. The XML framing protocol is a logical and human readable

way to identify data fields and structures.

Most devices being converted to Internet appliances already have their human interface; it is the front panel. Some of the more complex devices will have an embedded kernel with its own built-in human-machine-interface (HMI). Some may not require an HMI or may use a Web browser to create the human interface. There are few, if any, automatic testers running as Internet appliances have the need to communicate with AOL, CNET, or Yahoo. These devices have a limited and unique requirement and that is to display specific data and control the device thus having little need for a Web browser. And, most embedded Web browsers have heavy memory requirements and processor loading.

There may be a need for HMI compatibility between the local and remote user panel. That might mandate the use of an embedded Web browser in the device. Such a need will be shown later. The embedded Web client, without a browser, is an underused Internet application. It has a very small memory footprint but with much of the power of a Web server. It communicates with Web servers by requesting or sending objects and with the embedded application. The memory requirement of an embedded Web client is about 12 KB mix of ROM and RAM.

There are few companies currently offering embedded Web clients Allegro Software Development has been shipping their WebClient software for a few months. It is being used in a telecommunications tester and a time-and-attendance terminal. Ebs, Inc. has had an embedded Web client on their price list for quite a while with the tag, TBD, in place of a price. The more common way to make a device into an Internet appliance is with the use of an embedded Web server. The embedded Web server communicates with other Web servers and with user-provided Web browsers.

```
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)) {
        AllegroTaskDelInit(
            theTaskDataPtr); theTaskDataPtr == (void *) 0;
    }
}
```

Figure 1. The RomPager server engine is invoked with just a few lines of code. The lines of code above can be used where there is no operating system in the device.

Web Server

The embedded Web server delivers HTML formatted graphic pages to Web browsers and communicates with other Web servers on the local network or the

Internet. These graphic pages can display the data normally seen on the front panel of the instrument and accept user input to control or setup the device. Most of the embedded Web servers can be incorporated with their HTML pages in under 100 KB of ROM and RAM.

What follows is the creation of an Internet appliance using an embedded Web server. The costs and memory sizes listed are based on the Allegro Software Development RomPager products. There are competitive offerings from companies as Treck, Prism, Spyglass, and others. Many RTOS vendors have offerings through alliances with the vendors of embedded Web servers. Most embedded Web servers, such as the Allegro RomPager, are provided as C source code, are process or and operating system independent, and are usually licensed without runtime royalties.

The key to using a Web browser to emulate the front panel is the ability to display dynamic or changing data on the remote browser. This capability is provided by the embedded Web servers. Realtime data is provided to the HTML pages through buffers, variables, and functions. The embedded Web server connects this realtime or dynamic data to the requested HTML page and the page with the data is sent to the remote Web browser. In order to alter the display of the dynamic data in the field, you will want to be able to change the pages on the fly. You need a runtime HTML page parser in the embedded Web server such as the Allegro SoftPages product.

Most embedded Web servers can be easily married to the embedded application used to control the instrument. An embedded kernel or realtime operating system (RTOS) may be used but is not necessary for most embedded Web servers. The RomPager embedded Web server engine when used without an RTOS is invoked with just a few lines of code (Figure 1).

Whether you need an RTOS or can run just as an embedded application is determined by the implementation of your instrument. An RTOS isn't required to run an embedded Web server. The Allegro toolkits come with make files for all popular RTOSs. If you do not use an embedded kernel or RTOS, the integration is quite simple. The code required for integration is in Figure 1.

Most embedded Internet toolkits come with make files for the popular embedded kernels, operating systems, and TCP/IP software stacks. Moreover, they are processor independent.

One can make the Internet appliance proactive by adding embedded e-mail. With embedded e-mail, the device can send a message to the responsible person when the device needs service. One can also use e-mail to download software updates and services. When using an embedded Web server in the Internet appliance, each of the end-users need a Web browser. Then, most already have them.

Using Web technology provides the Internet appliance with a common network interface, a recognized user interface, and a connection to any platform or user. It is inexpensive to develop and one does not have to be an Internet guru to make it happen.

LOW DEVELOPMENT COST

Web technology software costs much less than expected. The software tools and application code to develop a very complete Web enabled device with e-mail will cost about the manufacturer \$50,000 - maybe less. Most embedded Web server software is licensed without run-time royalty fees. This \$50,000 should be compared with the cost to implement the Web functionality in-house. It is estimated it would take two engineers about four months to get a reasonable embedded web server running but without support for all the RTOSs and TCP/IP stacks. Fully loaded costs for an engineer is about \$150,000 per year making the four month effort cost about \$100,000. Then they still have to integrate the Web server into their product. Buying an Internet toolkit makes this effort a lot easier and faster. Some of the toolkit vendors have hundreds of thousands of their embedded products running in devices. These commercial toolkits have been well debugged and are maintained to current Internet standards. This is difficult to duplicate-house.

One of the best parts of using Web technology software, there is no user software to distribute, no new software for a different connected platform, and no tracking of users. The HTML graphics and data displayed on the remote browsers are stored in the Internet appliance and sent to the user via the network, on demand. The user provides the remote browser and computer. The HTML graphics can be displayed on any commercial browser. The embedded Web server will provide some security (usually logon/password) and the corporation's firewall will keep outsiders away. Probably all of your customers are familiar with the use of a Web browser so training of end users is limited to educating them about the tester.

Web Server Options

Embedded Web servers come in versions from minimal servers to full-featured servers. The amount of memory required by the server will vary according to feature sets. Most projects will probably migrate to a more full featured server. The additional cost and hardware support is not great.

E-mail is an option not usually considered for use in a machine controller. The Web server vendors can elaborate on the possibilities and the advantages gained from having e-mail added. A feature gained from the inclusion of embedded email is the ability of the test equipment to send an email alert to a supervisor. Another feature gained is to be able to update the internal software by attaching the updated code to an email message sent to the test equipment.

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

Allegro Software Development offers a runtime HTML parser, SoftPages, to permit end users to design their own front panel for the Internet appliance. Customers can create the display that meets their needs and customize it with their own logo. All the vendor does is pro-

Product Name	Description	ROM	RAM	Standards Supported
RomPager Advanced	Embedded Web Server, full featured, communicates with all Web browsers. Includes Web Application toolkit with compile-time HTML parsing and compression	13-29Kb	3-5Kb RAM per request	HTTP 1.0/1.1, HTML 2.0, 3.2, 4.0, RFC 1867, RFC 1945, RFC 2068, RFC 2069, RFC 2616, RFC 2617
RomPager Basic	Embedded Web Server, small footprint, communicates with all Web browsers	7-12Kb	2-4Kb RAM per request	HTTP 1.0/1.1, RFC 1945, RFC 2068, RFC 2616
RomPager SoftPages (RomPager option)	Runtime HTML parser. Permits users to create their own GUI.	15Kb	5Kb	HTML 2.0, 3.2, 4.0
RomWebClient	Embedded HTTP client, sends and receives to all Web servers	10Kb	2Kb	HTTP 1.0/1.1, RFC 1945, RFC 2068, RFC 2616
RomMailer	SMTP Client, sends Internet e-mail. Often used to send notices to supervisors of an alarm or the need for service.	4-8Kb	3Kb	SMTP, MIME attachments, RFC 821, RFC 822, RFC 2045, RFC 2046, RFC 2110, RFC 2111, RFC 2112
RomPOP	POP3 Client, receives Internet e-mail. Can be used to update internal software.	12-18Kb	10-20Kb	POP3, MIME attachments, RFC 822, RFC 1939, RFC 2045, RFC 2046
RomDNS	Domain Name Services client	3Kb	5-20Kb	RFC 1034, RFC 1035
RomPager Java Graphlets	Java applets for dynamic data presentation	15-100Kb	0Kb	Most popular browsers
RomXML	Embedded XML parser/framer	10Kb	3Kb	W3C XML

Figure 2. Allegro Software Development RomPager Embedded Internet Toolkit family of software. RFCs (Request for Comments) are the standards used to identify the various parts of the Internet.

vide the customer with the variable names and formats that are used inside the instrument. Having a runtime HTML parser could provide the instrument manufacturer a new source of revenue - writing custom pages for their customers.

The Allegro RomXML parser/framer can give the appliance a state-of-the-art method of sending and receiving data. It can parse XML formatted data and pass the information into C programming constructs.

Any of these functions can be used individually or collectively. Web technology is truly a menu of features that one can select any combination and be reasonably sure their choices will cooperate.

Some of the vendors with embedded Internet products include:

- **Allegro Software Development** provides an advanced and basic embedded Web server, embedded Web client, embedded email, XML parser/framer, and HTML runtime parser.
- **Treck** sells both the TCP/IP stack (optimized for embedded and realtime applications) as well as the embedded Web server.
- **Phar Lap Software** has a proprietary Win32 RTOS, TCP/IP stack, and embedded Web server.

- **Spyglass** ships both an embedded Web server and embedded Web browser.

- **Ebs** has a minimal embedded Web server, TCP/IP stack, and a newly announced embedded browser.

Go to the Internet World's WebCompare Web site for a partial list and comparison of available Web servers. (URLs are provided at the end of the article.)

Figure 2 is an overview of the Internet functions that can be added and their memory requirements. I am using products supplied by Allegro Software Development but they could come from other vendors with similar parameters.

Internal Computing Power

Most instruments and controllers that are candidates to become Internet appliances are complex and are based on a microcomputer chip. Whether that chip is an eight-bit, 16-bit, or one more powerful, is not a problem. If the microcomputer chip can support the network protocol, it can support the Web technology software that will make the instrument into an Internet appliance. Some vendors have used the 8051 microcontroller but more typical is an 80186 or one of the many 32-bit processor chips. The TCP/IP software vendors will say whether their software will run on the microprocessor chip used in the instrument.

For the instrument conversions proposed later in this paper, it is assumed there is a powerful enough micro-computer and is the necessary network interface hardware exists. The network hardware can be either a serial port or an Ethernet connection. A serial port is less desirable. It requires the instrument to have some sort of external modem or an external network server.

If one was designing an instrument from scratch, one might consider using microprocessors from companies like Connect One or NetSilicon. These chip vendors have included some of the network and Web protocols as part of their microprocessors.

Network Interface

If the instrument is equipped with a serial port, it could be used to 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. One of the earliest small portable Web servers came from Dawning Technologies, Inc. of Fairport, NY. As with most external servers the instrument manufacturer must write software that communicates with the instrument through the serial port as well as designing the pages that are used to control and monitor the instrument.

The advantage of this solution is the instrument manufacturer does not have to modify the existing product to make it into an Internet device. The disadvantages are the added cost of external hardware and the user is limited to what existing functionality is available through the serial port.

The hardware cost difference is similar to the cost difference between an internal and external data/fax modem for a PC. This is a good solution for the end user if the instrument has no network connection but has some functionality available through the serial port. The solution is limited to the functions provided through the serial port.

Most instruments designed in the past few years have a network connection, usually Ethernet and TCP/IP. The idea for these instruments is the data can be sent to external computers for further processing and storage. What usually cannot be done is setup and control of the instrument across the network.

With basic network functions built in, the only piece missing to become an Internet appliance is the embedded Web server. One may need additional ROM and RAM but I would bet that most instruments have a surplus of memory or the ability to exchange the memory chips with those of greater capacity without a redesign.

WEB SERVER INSIDE

With a Web server inside the instrument, one can design HTML graphics pages that look like the original hardware. Rotary knobs might be replaced with sliders and toggle switches with clickable push buttons. CRT-like displays and analog meters can be replicated using Java or ActiveX driven graphics. The entire human interface can be displayed remotely through a Web browser. The pages for the Web browser can be

created using tools as FrontPage or Dreamweaver HTML page generation software.

The instrument manufacturer will have to write code to communicate between the Web server and the various functions of the instrument. This consists of pointers to buffers, variables, and functions. The inclusion of the dynamic data in the HTML pages is the insertion of the variable and function names in the HTML code. This is accomplished in different ways by the various vendors. Most have a method to identify the dynamic data variables different from the HTML code.

When the Web server and HTML pages are complete, all of the front panel functions, controls, and displays can be made available to the operator's browser anywhere on the network where network access to the instrument is available. That might be on the bench along side the test equipment or in an office two buildings or two countries away. Then quality control manager in his office can monitor test results in realtime while the parts are being tested on the production line.

REMOVE THE FRONT PANEL

Once one has an Internet connection to the instrument and the users have access to all front panel functions through their PCs, why keep the front panel? Front panels are expensive to build and are the responsibility of the manufacturer to support. If the front panel can be replicated by HTML graphic pages on a Web browser, all that may be needed on the front panel is an ON-OFF switch.

Looking at the typical instrument one usually finds a

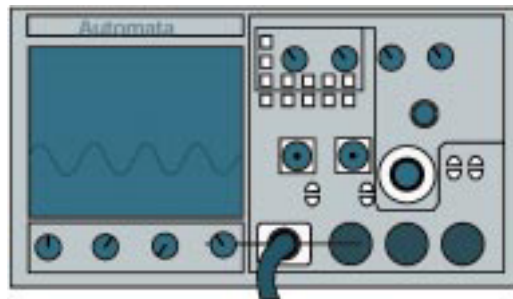


Figure 3. Original instrument, the basis for a new Internet appliance.

set of knobs, switches, and meters. These knobs and switches are used to set up the device so it can measure or monitor external signals. They are used to set up the method of display of the monitored values. A front panel full of controls and displays are expensive to manufacture. They contain electromechanical devices that are usually hand wired and contain expensive panel displays. The cost of the front panel for many instruments is half the cost of the entire instrument. In many cases, the power supply of the instrument can be slimmed down if it no longer must support a power hungry video display.

Making an instrument into an Internet appliance can give it added functionality at a fraction of the cost of the

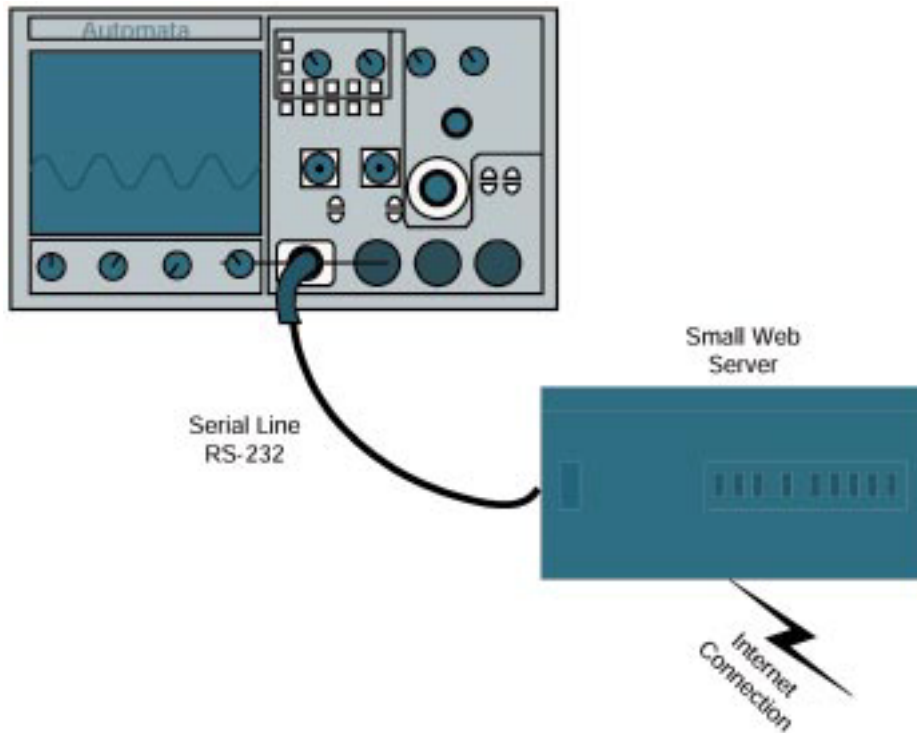


Figure 4. A simple way to convert existing devices into Internet appliances is with the use of an external Web server. The instrument must have a port available with setup and data collection functions.

original instrument. Customers now provide the front panel through their Web browser and they select the type of computer best suited for their needs. That front panel can be wherever the user wants to work. With the functions available to users via the Internet, there can be multiple users monitoring the instrument with usually one user in control.

Access to HTML pages with control functions can be

limited by either user discipline or the manufacturer can include security features usually provided by the Web server toolkit. Primary protection against outside intruders is provided by the corporate firewall.

CREATE FIVE PRODUCTS WITH COMMON COMPONENTS

One can take a test instrument and turn it into an Internet appliance. Using the same basic chassis, electronics, and internal software, at least three and possibly five different products can be created. This use of common components makes the Internet appliance an attractive consideration. Starting with a stand-alone machine controller or instrument (See Figure 3) add the processing power, extra memory, network hardware and software interface, and any one of the various Internet applications used to create an Internet appliance. Using this as the base appliance, five different products that will be created in this article.

One can keep the original product or upgrade it with a higher speed processor, network interface, and additional memory. Then one can include the Internet software and not activate it. This product now becomes the basis for the four follow-on products that are Internet appliances. One option for the manufacturer is to sell this as the original product was sold but with the networking turned off.

- 1 The original unmodified instrument can be used as an Internet appliance if it is equipped with a serial port that has the setup functions and output data accessible through it. The serial port can be connected to an external Web server and through

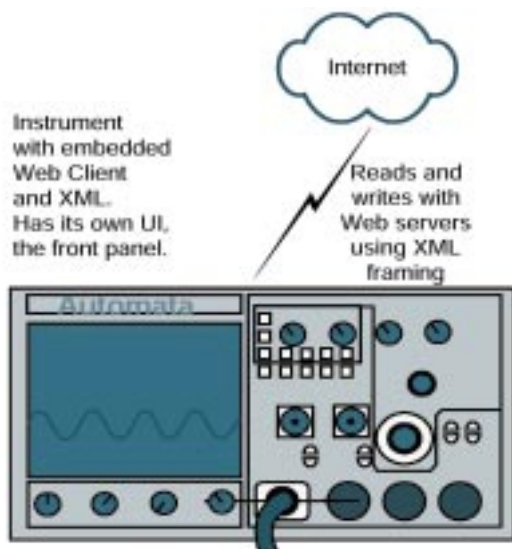


Figure 5. Some instruments need communications with information servers. Using the embedded Web client will provide easy to use, two-way, communications.

AD FORCE

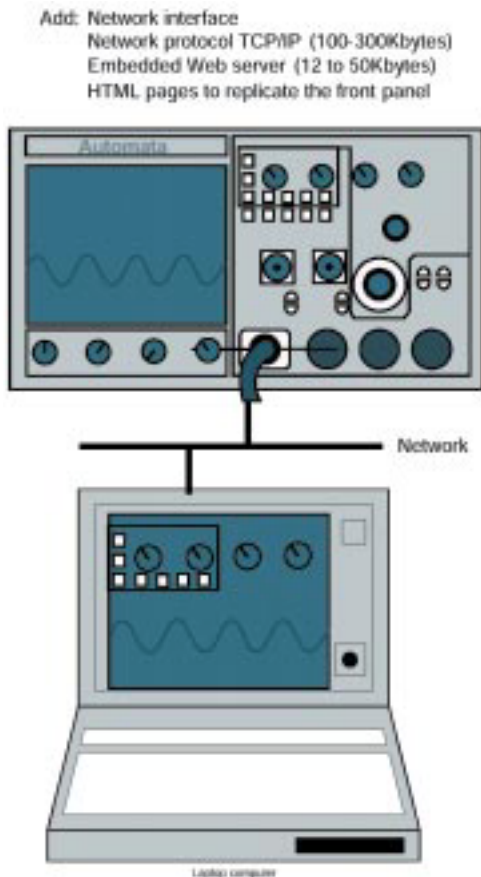


Figure 6. The most popular configuration of an Internet appliance is to add an embedded Web server so the front panel can be replicated on a remote PC using a Web browser.

this Web server the instrument is accessible anywhere on the network using a Web browser. This option is usually used with older instruments or users may connect existing devices to the network in this fashion. (See Figure 4)

2. Now add the network interface and software stack plus an embedded Web client. With the embedded Web client, the appliance easily communicates with information servers to download setup commands and to upload data. The XML parser/framer software can be added to make the information easier to search and utilize. The XML software can be used in any of the Internet appliance configurations. (See Figure 5)
3. Instead of adding the simple client software, add the more robust embedded Web server software to allow the use of remote Web browsers to display and control data to and from the appliance. Users can use their desktop PC, laptop, or any Web terminal with a full-featured browser to access the Internet-applianced instrument. The front panel functions can be replicated using HTML pages with dynamic data capability provided by the embedded Web server. From the user's remote Web browser

the Internet appliance can be managed, setup, and its data displayed. Graphical data display imitating CRTs, analog meters, or chart paper can be created using Java applets usually provided by the embedded Web server vendor. (See Figure 6)

4. To produce a cost reduced version of the Internet appliance remove the front panel and only use the remote browser to replicate the expensive front panel. Often the front panel of a tester is the most expensive portion. The panel contains electro-mechanical components, CRT or LCD display, and many rotary and toggle switches. These can all be replicated in software and distributed as HTML pages. Replacing the front panel with a user supplied PC and Web browser could cut the cost of manufacturing the Internet appliance in half. (See Figure 7)
5. Finally, a more expensive option is to use an embedded Web browser inside the Internet appliance and use the same HTML pages to display on the front panel as well as on the remote browsers. The existing electromechanical panel is replaced with something similar to a laptop computer that can run a full-featured Web browser. The look and feel of the front panel will be identical on both the local appliance as well as on any remote browser accessing the Internet appliance across the network. (See Figure 8)

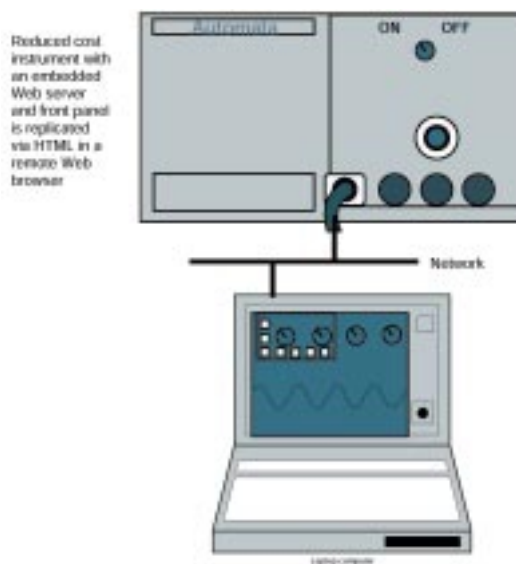


Figure 7. Removing the front panel from an Internet appliance is an excellent cost reduction. The front panel now becomes a user supplied laptop and Web browser. All the panel functions are replicated with HTML pages and Java or ActiveX graphics.

The use of common components can reduce the cost of the Internet appliance permitting easier manufacturing, stocking of fewer parts, and less service. The products would all look similar and have the same user interface.

Five Products from One

With Web technology in your machine controller or instrument, you might have five unique products all with nearly identical electronics and software. (See Figure 9) The basic unit can be designed with the appropriate memory and software support but without the external network connection. This product will look identical to the original instrument. Adding the external network connection and turning on the web server software, you now have an Internet device with lots of additional functionality. Finally, you can remove the expensive components on the front panel, install a new front panel, and have a cost reduced product with the same functionality of the previous product.

Adding such options as the Allegro SoftPages and e-mail can be done through the Web server software and provide you with a follow-on revenue stream. SoftPages permits your customers to modify the look and feel of the system and enhance it for their unique installation. You could even provide a service where you add their logo and new functions for additional revenue. Turning automatic test equipment into an Internet device should increase market size and definitely increase sales and margins. All this and at almost no cost.

Not only does the use of Web technology make sense financially; the Web browser is becoming a default user interface for more than just AOL. All software now resides in the device thus no distribution issues or the possibility of illegal copying.

And it is SEXY to use Web technology in devices.

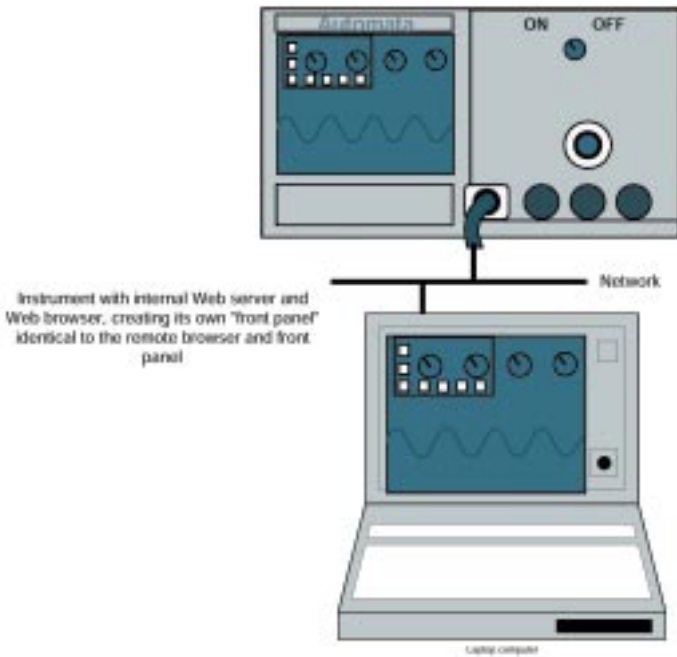


Figure 8. Often it is desired to have identical user interfaces both on the Internet appliance as well as on the remote Web browser. An embedded Web browser and laptop-like front panel can achieve this at a high cost.

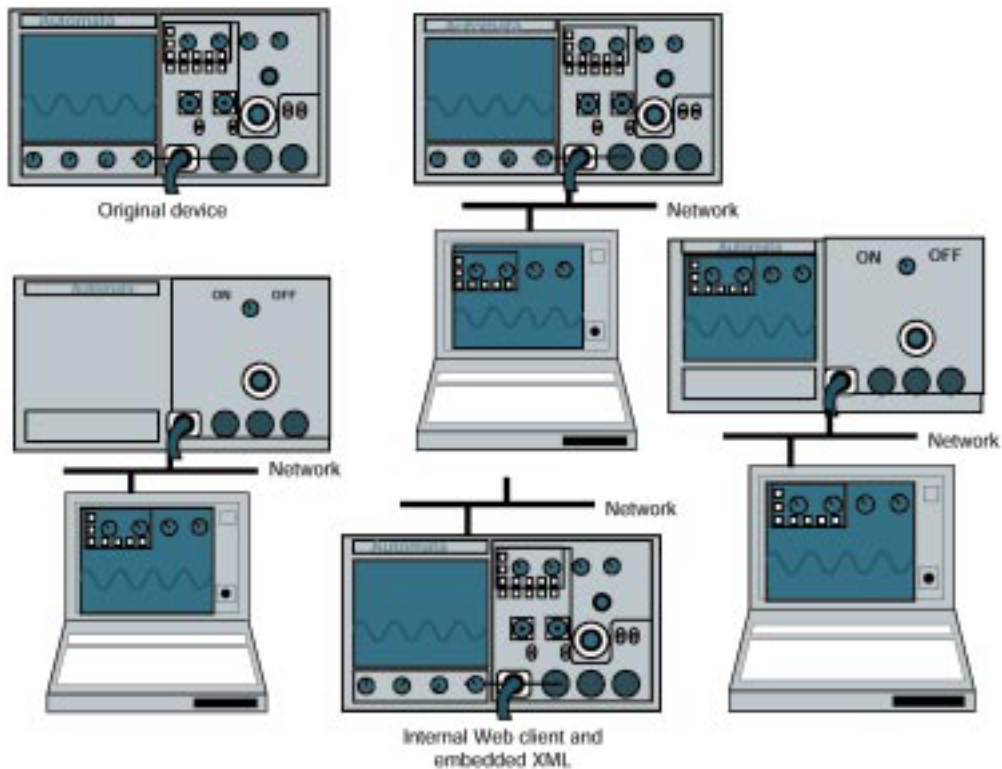


Figure 9. From a basic design, five unique products can be created to form a family of Internet appliances.

REFERENCE WEB SITES

The following URLs are for companies mentioned in the text.

- webcompare.internet.com - Internet World's Web comparison site.
- www.allegrosoft.com - Allegro Software Development Corporation
- www.connectone.com - Connect One Ltd.
- www.dawning.com - Dawning Technologies
- www.etcbin.com - Ebs, Inc.
- www.netsilicon.com - NetSilicon, Inc.
- www.pharlap.com - Phar Lap Software, inc.
- www.spyglass.com - Spyglass
- www.treck.com - Treck, Inc. ■

Edward F. Steinfeld is an Embedded Computing Market Consultant at Automata International Marketing.

Ed has more than 25 years experience in realtime and embedded computing. He began as a programmer writing code and designing hardware to test hybrid circuit boards for Picker X-ray. He has marketed embedded and realtime products to OEMs and resellers for Digital Equipment Corporation, VenturCom, Inc., and Phar Lap Software. His international experience includes a stint in Hong Kong as a Far East Channels Manager and responsibility for international OEM sales in Europe and the Pacific Rim. Ed is now providing market research, planning, and services to the embedded computing industry.