ETHERNET invades embedded space

WARREN WEBB, TECHNICAL EDITOR

Fueled by widespread use on the desktop, new low-cost hardware, and widespread demand for connectivity, Ethernet is gaining ground as the network of choice for embedded-system developers.

You live in an information-hungry age. You need instantaneous access to every piece of information that defines how your organization ticks. The enterprise Ethernet network brings administrative, financial, and engineering data to the desktop, but, more likely than not, many of your embedded factory-automation, test, and field systems operate independently. With the availability of low-cost hardware and off-the-shelf software, it is time to extend Ethernet networking to the lowest company level. The more systems that are connected, the faster you can react to problems or opportunities.

Fast access to information can turn big problems into routine tasks. For example, avionics manufacturers must do a complete analysis of certain in-flight failures before the airline recertifies the product. Engineers need access to manufacturing records to see what parts went into each serial-number unit. Test-equipment records show whether there were any anomalies in temperature cycling or final test. Software-integration test results verify proper software testing. Data from design reviews show engineering analysis of the circuits in question. Operational-history records reveal whether there were any related prior failures or similar failures on other units. Manually gathering this much data may take weeks and may require several team members, but a good companywide network can reduce the problem to a few hours of work.

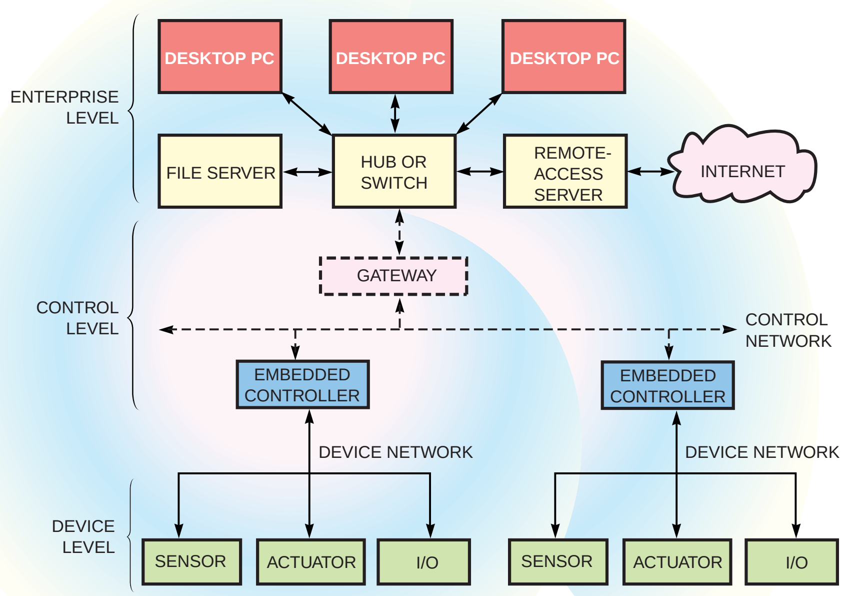


FIGURE 1

Embedded Ethernet offers a low-cost method for connecting the enterprise-level network to the control and device levels.

Figure 1 shows a simplified diagram of the information network in a typical manufacturing organization. Ethernet is the most popular network for the enterprise level, with predictions of 190 million ports in use by 2000. However, network choices are less defined when you drop to the control level. In fact, most embedded factory-automation controllers are stand-alone or are connected by proprietary serial communications.

When engineers connect embedded systems, they generally select serial communications because it is the best understood, easiest to write interface software and is available in every μP. Designers base most embedded-system serial communications on RS-232C or RS-422/485. RS-232C, the serial port built into most μPs, is a full-duplex, point-to-point link allowing communications at data rates as high as 115 kbps (although slower rates are far more reliable). Designers generally use the RS-232C port as a local user-interface port because it is limited to distances of 50 ft and to one node.

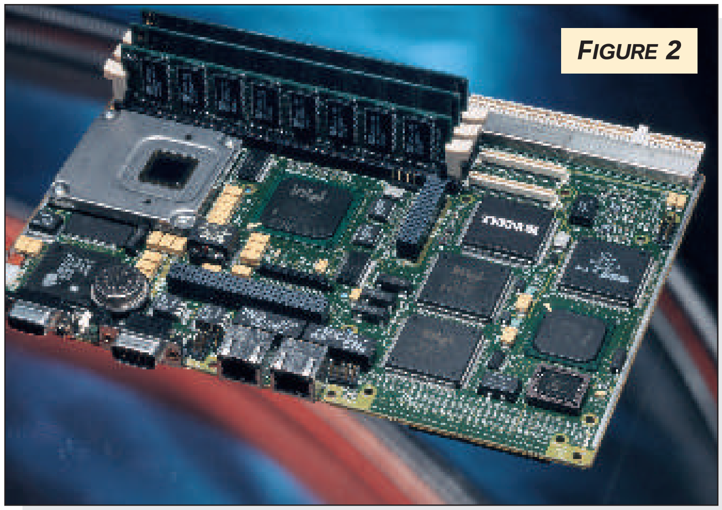


FIGURE 2

Teknor's CPCI1003 CompactPCI single-board computer ($2795) includes two Ethernet ports to minimize network traffic on the control level.

Multidrop problems

Unlike RS-232C, RS-422 (four wire, full duplex) and RS-485 (two wire, half duplex) use differential line drivers and receivers to increase node separation to 4000 ft. RS-485 is the most popular approach because it can function as a multidrop network with as many as 32 nodes. However, the RS-485 specification defines no standard protocol for communicating over the network. Many users have developed their own techniques for RS-485 multidrop communications but have found problems both with identifying the master node and with rapidly switching the data direction without dropping bits.

In spite of these problems, developers continue to select and use RS-485 communications hardware. Over the past 10 years, vendors have developed several hundred proprietary-control-bus architectures based on RS-485 multidrop or similar physical hardware. Each architecture claims to solve some unique environmental, data-volume, timing, or network-media problem. Engineers developed controller-area network and J1850 standards to reduce automotive wiring and LonWorks, CEBus (consumer-electronics bus), and BACnet (building-automation-and-control network) for building automation. These proprietary systems require specialized hardware and software, which is available from one or a few vendors. Proprietary networks have higher per-node hardware and software costs but lower multidrop-wiring costs than Ethernet.

@ a glance

- Ethernet beats proprietary serial communications as the preferred embedded-system network.
- The Internet and Ethernet team up to give worldwide remote access to embedded products.
- You can substitute Web browsers for user-interface hardware to simplify networked embedded systems.
- Networking positions your product for information interaction with customers, engineering, market research, and management.
- Ethernet offers several advantages over proprietary networks, including low-cost hardware, open standards, off-the-shelf software, and higher speeds.

Frustrated by the lack of a standard network at the control level, many engineers are turning to Ethernet. Although it is not a perfect fit for embedded systems, Ethernet has a number of advantages over proprietary control buses. Ethernet is an open standard with many vendors competing to produce the cheapest and most flexible products. Many organizations have Ethernet experts from the enterprise network. Ethernet also provides a growth path to higher performance and worldwide access over the Internet.

Extending the Ethernet network into embedded space requires the same topology and media that the enterprise uses. 10BaseT (10-Mbps) and 100BaseT (100-Mbps) Ethernet use a star topology with a centralized hub. The primary advantage of the star topology is system reliability. If a segment of the network fails or breaks, such a breakdown affects only that link. The other portions of the network continue to operate normally. The most popular wiring medium for Ethernet is unshielded-twisted-pair (UTP), which is available in a variety of grades. Category 5 has the highest performance and supports 100BaseT Ethernet. Therefore, developers usually select Category 5 UTP even if the current application is 10BaseT.

TCP/IP fits all

In addition to selecting the topology and medium, embedded designers must select the network protocol. If you expect to connect to the enterprise network and have access over the Internet, then your choice is Transmission Control Protocol/Internet Protocol (TCP/IP). Designers working on the Internet developed TCP/IP and placed it in the public domain. Because it was an open standard and easy to implement, TCP/IP acceptance was widespread (see **sidebar** "The OSI reference model from the ISO"). Software companies now offer optimized TCP/IP software for almost every embedded μP supported by a real-time operating system (RTOS).

However, Ethernet coupled with TCP/IP is not perfect. Ethernet is a shared-media interface with rules to handle simultaneous data-transfer requests without dropping data. A collision results when two Ethernet nodes attempt to transmit at the same time. Nodes must then wait a random time interval and attempt to retransmit the data. As the amount of traffic increases on the network, the number of collisions increases. Because a node may have to retransmit data and because the time required to reach the destination is variable, Ethernet networks are nondeterministic.

Proprietary-network developers point to determinism as a major operational issue at the control level. However, engineers that use Ethernet argue that there are ways to deal with timing issues. At 10 Mbps, Ethernet is considerably faster than most proprietary buses so that, even with collisions, network speed is not a problem. Nevertheless, when speed is critical, 100-Mbps Fast Ethernet (IEEE standard 802.3u) is available, and 1000-Mbps Gigabit Ethernet is on the horizon.

FIGURE 3

The new 3.6×3.8-in. PC/104 single-board computer from Micro/sys includes an Ethernet port and TCP/IP software preloaded into flash memory.

Another technique engineers use to live with a nondeterministic network is reducing the traffic on the network segment. An intelligent switch in place of a hub divides networks into smaller, faster segments by examining data packets and forwarding only those with the proper address (**Figure 1**). Reduced network traffic results in fewer collisions. Segmenting the network also helps when operation must continue even with hardware failures.

Newer embedded single-board computers also include multiple Ethernet interfaces to reduce network traffic. One interface connects to the enterprise network, and the other connects to the embedded control network. Teknor Industrial Computers and General MicroSystems (www.gms4vme.com) offer CompactPCI single-board computers with two Ethernet interfaces (**Figure 2**). The computer acts as the Ethernet switch and filters traffic to the control side.

Determinism is not the only problem with Ethernet as an embedded control network. Opponents argue that Ethernet is a data network, not a control network. Some control environments use data packets as small as 10 to 20 bytes in which Ethernet overhead would be inefficient. If your proposed application has many small control messages, such as a real-time control loop, you may be better off with a specialized network optimized for the application.

Many control-level networks must also work in harsh or noisy environments, such as on the factory floor. Specialized proprietary buses tackle noise by slowing data transfer, increasing error checking, and changing the net-

THE OSI REFERENCE MODEL FROM THE ISO

Open System Interconnection (OSI) is an international standard that defines a seven-layer model for implementing network protocols. The International Standards Organization (ISO), the same group that introduced the ISO9000, defined the OSI model. Although the functionality of the seven-layer model exists in most communications systems, it is too loosely defined to serve as a design specification. Today, the ISO model serves as a framework to define other protocols. **Table A** defines the seven layers of the OSI reference model.

TABLE A–THE SEVEN LAYERS OF THE OSI REFERENCE MODEL

Layer	Name	Function	TCP/IP example
7	Application	Application communications interface	E-mail, file transfer
6	Presentation	Data conversion (for example, EBCDIC to ASCII)	Not used
5	Session	Application synchronization of communications channels	Post-office Protocol, File Transfer Protocol
4	Transport	Defines how connections are established	TCP
3	Network	Routes data from one node to another	IP
2	Data link	Defines frame format, encoding, and error checks	Ethernet interface driver
1	Physical	Provides characteristics of the node hardware	Network interface

Note: EBCDIC=extended binary-coded decimal-interchange code.

work medium. Both proprietary buses and Ethernet (10BaseFL) use fiber optics in areas of increased EMI.

Worldwide connectivity

You might want to consider Ethernet for worldwide connectivity, if you can live with the data rates and environmental constraints. Imagine the advantages of sitting down to any networked PC and calling up one of your products to evaluate operation. You could update firmware or diagnose and troubleshoot problems. Except for probing with external test equipment, you have all the information that you would have if you were on-site.

Remote access is one of the most exciting features of a fully networked embedded system. By incorporating a Web server into your Ethernet-connected embedded product, you can eliminate the built-in keyboard and display and create a portable graphical user interface from any networked computer. A Web server is simply a program that waits for a TCP/IP connection with a client program, such as a Web browser. Once the software makes the connection, the server delivers a predefined data stream, such as an HTML page (**Reference 1**).

Your remotely installed and networked embedded product has a host of advantages over a non-networked product. You can direct the initial product delivery and installation at a remote site over the Internet. Factory engineers can monitor critical data during operation and can even remotely recalibrate the product. You can program remotely networked products to automatically self-monitor and forward e-mail to selected individuals if an error occurs. The embedded system can also receive an e-mail message to perform a specified action, such as self-test. You can temporarily download large test diagnostics into flash memory or RAM for execution, deliver program updates or even new software products over the Internet, and gather statistical information about fielded units for engineering and marketing.

Many designers build embedded systems, such as test equipment and industrial controls, from ruggedized, desktop-compatible hardware and software. This segment of the embedded-system market continues to grow because of the low-cost hardware and vast amount of software. In this case, adding Ethernet connectivity is simple: Just plug in an off-the-shelf network card and configure the operating system (probably Windows 9x or NT). Your application can then send and receive information over the network by communicating directly with the Windows application-programming interface or through such devices as Visual Basic's Winsock ActiveX control.

Vendors want to take some of the mystery out of networked-embedded-system development by providing hardware with ready-to-run, onboard Ethernet software. Micro/sys just released a PC/104 single-board computer, the SBC1190 (**Figure 3**), which includes a 10BaseT Ethernet port and 48 analog and digital I/O lines. Micro/sys also preloads its TCP/IP software in flash memory so designers can easily implement network communications through higher level Winsock calls. The company includes sample Windows NT programs to exchange data with remote

FOR MORE INFORMATION...

For more information on products such as those described in this article, circle the appropriate numbers on the Information Retrieval Service card, or use *EDN*'s Express Request service. When you contact any of the following vendors directly, please let them know you read about their products in *EDN*.

Advanced RISC Machines
Los Gatos, CA
1-408-399-5199
fax 1-408-399-8854
www.arm.com
Circle No. 308

Cirrus Logic
Fremont, CA
1-512-912-3299
www.cirrus.com
Circle No. 309

Hewlett-Packard
Palo Alto, CA
1-800-452-4844
www.tmo.hp.com
Circle No. 310

Integrated Systems Inc
Sunnyvale, CA
1-408-452-1500
www.isi.com
Circle No. 311

Micro/sys
Glendale, CA
1-818-244-4600
fax 1-818-244-4246
www.embeddedsys.com
Circle No. 312

Motorola Semiconductor Products Sector
Austin, TX
1-512-895-3823
fax 1-602-244-6560
www.mot-sps.com
Circle No. 313

Osicom Technologies Inc
Santa Monica, CA
1-310-581-4030
fax 1-310-581-4032
www.osicom.com
Circle No. 314

Teknor Industrial Computers
Boca Raton, FL
1-561-883-6191
fax 1-561-883-6690
www.teknor.com
Circle No. 315

VOTE . . .
Please also use the Information Retrieval Service card to rate this article (circle one):
High Interest 590
Medium Interest 591
Low Interest 592

Super Circle Number

For more information on the products available from all of the vendors listed in this box, circle one number on the reader service card. **Circle No. 316**

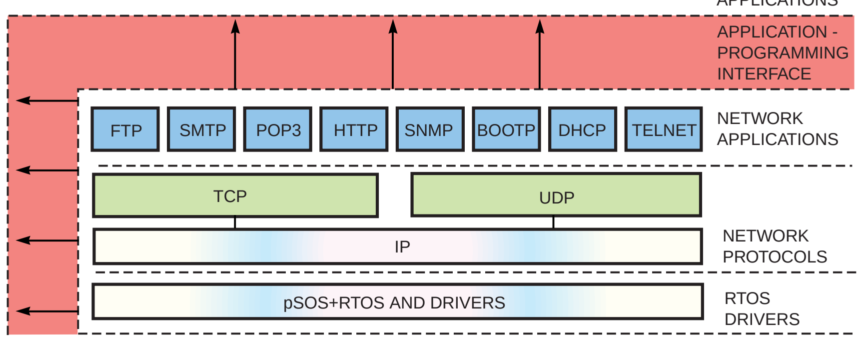
ETHERNET FOR EMBEDDED SYSTEMS

SBC1190 computers. The single-unit price for the SBC1190 with Ethernet is $355.

Several vendors provide Ethernet-controller chips, which you can design into embedded products. Cirrus Logic's CS8900 CrystlLan embedded Ethernet controller supports 10BaseT Ethernet and costs $9.95 (1000); a Windows CE driver for the CS8900 is free. Motorola's MC68160 enhanced Ethernet serial transceiver, an integrated serial-interface adapter and transceiver, is available for $15.48 (160).

Osicom Technologies promises to take networking to the device level by delivering the mP, the 10/100BaseT Ethernet, the RTOS, the communications software, and all software licenses with a single ASIC that sells for $32.50 (10,000). Osicom's chip, the NET+ARM, and the associated software integrate a 32-bit ARM7TDMI RISC processor core from Advanced RISC Machines (**Reference 2**), the pSOS+ RTOS from Integrated Systems, and Ethernet. User-application programs have direct access to a complete communications-software suite (**Figure 4**) and to the RTOS. Osicom also offers a NET+ARM development kit ($20,000) that includes a reference design, development tools, training, and support.

FIGURE 4



NOTE: BOOTP=BOOTSTRAP PROTOCOL.

The NET+ARM ASIC from Osicom includes software for Ethernet file transfer, e-mail, and network addressing, along with Integrated Systems' pSOS+ RTOS.

The Osicom ASIC is a good example of the hardware and software necessary to implement an embedded-system networked product (**Figure 4**). You need a fast processor to execute both the application and the network software. An RTOS ensures that the network software and the application software share processor time with appropriate priorities. An Ethernet hardware driver interacts with the physical network circuitry. TCP/IP software implements the protocols that the IEEE standard requires.

Ethernet ports are beginning to show up on test instruments in which serial or IEEE 488 interfaces are the rule. The Hewlett-Packard Infinium oscilloscope family now includes a standard 10-Mbyte Ethernet port. The network connection allows remote setup, data transfer, and remote printing to network printers. Engineers can transfer data and instrument setups directly between their desktop PC and the lab or factory.

Ethernet and TCP/IP are rapidly becoming the standard communication techniques for many embedded systems. Ethernet can enhance your embedded design by eliminating the local user interface, reducing the cost of field support, and improving customer interaction. Although TCP/IP requires more processing power and memory than traditional serial communications, the increase in customer benefits more than pays for any extra hardware costs. In the past, when embedded-development budgets dictated the use of a small mP with limited memory, software designers found TCP/IP difficult to implement. However, today, with some 32-bit mPs selling for less than $15 and memory prices at their all-time lows, you should consider Ethernet communications even for the smallest embedded products.

References

1. Quinnell, Richard, "Web servers in embedded systems enhance user interaction," *EDN*, April 10, 1997, pg 61.

2. Levy, Markus, "*EDN*'s 24th Annual mP/mC Directory," *EDN*, Sept 25, 1997, pg 34.

3. Shear, David, "Putting an embedded system on the Internet," *EDN*, Sept 12, 1997, pg 36.

You can reach Technical Editor Warren Webb at 1-619-513-3713, fax 1-619-486-3646, wwwebb@cts.com.

VOTE

Please use the Information Retrieval Service card to rate this article (circle one):

High Interest	Medium Interest	Low Interest
590	591	592