# Embedded Communication Protocol Options

**Bhargav P. Upender**
United Technologies Research Center
East Hartford, Conn.

Bhargav Upender is an assistant engineer at United Technologies Research Center. His research interests include protocol development, modeling and simulations, and performance analysis. He holds a BSEE from the University of Connecticut and an MEng in electrical engineering from Cornell University.

**Phil Koopman**
United Technologies Research Center
East Hartford, Conn.

Phil Koopman is a senior researcher at United Technologies Research Center. He currently designs and evaluates architectures and communications protocols for Otis elevators, Norden radars, UT automotive components, and other embedded applications. He has previously worked for Harris Semiconductors as an embedded CPU architect and the U.S. Navy as a submarine officer. Koopman holds a BSEE and an MEng from Rensselaer Polytechnic Institute and a PhD in computer engineering from Carnegie Mellon University.

## ABSTRACT

*Developers are realizing that traditional low-speed, point-to-point links are inadequate for their increasingly complex distributed embedded applications. Consequently, they are investigating multiplexed communication network protocols to incorporate advanced system capabilities, increase reliability, and reduce wiring requirements. This paper discusses special considerations for embedded system networks, a family tree of "standard" protocols, media access tradeoffs, and attractive options for off-the-shelf solutions. Based on real-time performance, cost, and hardware availability,* ***ARCnet, CAN,*** *and* ***LON*** *are strong contenders for most embedded systems.*

Embedded systems are becoming more and more complex. One of the ways to manage this complexity is to distribute the system functionality across several low cost microprocessors which communicate via a shared medium.

In the past, most physically distributed embedded systems used simple point-to-point links to provide inter-processor communication. With increasing demand for advanced features and the resulting drive for more flexible and cost-effective communications, engineers are starting to use LAN (Local Area Network) technology in embedded systems. Most LANs are based on Ethernet, which is ideal for workstation-like applications having aperiodic, bursty communication traffic. Unfortunately, many embedded systems are unlike workstations in that their communication networks must efficiently support periodic traffic, real-time constraints, prioritized messages, and cost-sensitive applications. In this paper we will discusses these special considerations for real-time embedded networks, explore "standard" protocols, discusses media access tradeoffs, and identify a few attractive off-the-shelf solutions.

## SPECIAL CONSIDERATIONS FOR EMBEDDED APPLICATIONS

Based on our examination of several embedded applications, we believe that communication traffic for embedded systems tends to be mostly short, periodic messages. Because cost limits the network bandwidth of many applications, protocol ***efficiency*** (message bits delivered compared to raw network bandwidth) is very important. Efficiency is improved by reducing packet overhead and media access overhead. Packet overhead is all non-data bits added by the protocol to ensure proper routing and reliable transportation (*e.g.*, CRC, address bits, acknowledgments). Media access overhead is the network bandwidth used to arbitrate network access among transmitting nodes (*e.g.*, token passing). Because worst-case behavior is usually important, efficiency should be evaluated both for light traffic as well as heavy traffic. For example, Ethernet is highly efficient for light traffic but gives poor performance if heavily loaded. Token passing protocols have the reverse properties. Therefore, protocol efficiency becomes a strong metric for selecting a protocol.

Due to real-time constraints of many control applications, ***determinacy***, the ability to predict message latency, becomes very important. Also, ***prioritization*** capability is required in some applications to allow quick channel access to critical messages (*e.g.*, safety critical conditions) and messages in which minimum latency is crucial (*e.g.*, sensitive control loops). Priorities can be either assigned to each node or to individual messages. Additionally, they can be either local or global. In local prioritization, each node is only aware of priorities of its messages, and arranges them in the transmit buffer accordingly. In global prioritization, the protocol allows the message or node with highest priority among all of the network to transmit.

Many applications require robust operation under extreme operating conditions. A protocol is ***robust***, if it can quickly detect and recover from errors (*e.g.*, duplicate or lost tokens). Some applications may require dynamic additions and deletions of nodes from the network. In these situations, the protocol should gracefully initialize and configure itself.

Varied operating environments may dictate use of a flexible ***physical layer*** that can support multiple media and mixed topologies. For example, a system may require expensive fiber in noisy environments, but can tolerate low-cost twisted pair wires in benign environments. Further, a bus topology may be optimum for wires, but a ring or star topology maybe needed for fiber.

Finally, the most important consideration is the ***cost per node***. Most of the protocols discussed in this paper are for high speed, high performance networks that allow expansion of the capabilities of a system (*e.g.*, remote monitoring, diagnostics, and servicing). Therefore, the current costs may not be suitable for low-end embedded systems. However, with the current trend of increasing computing power and protocol support embedded in CPU chips, the costs are becoming more reasonable for all types of applications.

## PROTOCOL FAMILY TREE

With the above considerations in mind, we surveyed the market for **standard** protocols for distributed applications. By identifying only standard protocols, we hoped to uncover low cost, off-the-shelf communication components and maintain interoperability with the other products. In particular, we hoped to discover one or two standards that were clear and obvious choices for embedded systems from both a technical and market perspective.

Much to our surprise, our survey resulted in more than sixty "standard" protocols. And, some of these standards specifically permit the use of multiple incompatible physical implementations. So much for simply picking "the" standard protocol for embedded applications!

In order to understand the relationship between these protocols, we developed a family tree (Fig. 1) for the most popular protocols. Most of these protocols can be well characterized as primarily addressing one of three different levels of standards.

- **Medium Access Control (MAC):** this level is part of the Data Link Layer of the Open Systems Interconnect (OSI) seven layer reference model[1]. This low-level sublayer defines the rules for bus sharing and arbitration. Every communication network uses one of these fundamental MAC protocols.
- **Protocol Implementations:** this level consists of hardware/software implementations of a MAC scheme. Market forces have made some of these protocols, the *de facto* standards in their application areas (*e.g.*, Ethernet, ARCnet).
- **High Level Standards:** this level represents protocols that are developed by world-wide standards committees. These standards are trying to provide cohesion and interoperability by addressing the higher, application layers of the OSI model.

## MEDIA ACCESS CONTROL MECHANISMS

In order to make sense of this tangle of standards, we will proceed from the low level to high level. MAC protocols determine the basic technical merits of any communication network. Once we understand each MAC scheme, we can then see how higher level standards fit them together.

### Connection Oriented Protocols

Before LANs became popular, connection-oriented protocols were heavily used to connect remote terminals to mainframes. Usually, the nodes are connected using point-to-point links (telephone wire, serial line, *etc.*). Communication between two nodes requires physical connection using handshaking signals, or logical connection via intermediate nodes. Connection

based protocols are deterministic between physically connected notes, and have readily available hardware and software. For an embedded system with modest communication requirements, this might be a cost effective protocol. Sometimes, this type of protocol is added to a more complex communication system to provide backward compatibility to older systems (*e.g.*, **BACnet**[2]). This type of protocol is used by the **X.25**[3] public network standard (network services offered by telephone companies) and IBM's System Network Architecture (SNA[3]).

### Polling

Polling is one of the more popular protocols for embedded systems because of its simplicity and determinacy. In this protocol, a centrally assigned master periodically polls the slave nodes for information. Since polling is done through some type of token (special string of bits) passing, this protocol is also known as the Master/Slave Token Passing or MS/TP. The majority of the protocol software is stored in the master and the communication work of slave nodes is minimal. This protocol is ideal for a centralized data acquisition system where peer-to-peer communication is not required. However, for a more complex embedded system, the single-point-of-failure from the master node is unacceptable. Additionally, the polling process has high MAC overhead and limited capabilities. These protocols have been standardized by the military (**MIL-STD-1553B**[4] and **MIL-STD-1773**[5]) for aircraft subsystem communications. Some variants of this protocol allow inter-slave communication through the master and multiple masters (e.g., **Profibus**[6]) for redundancy.

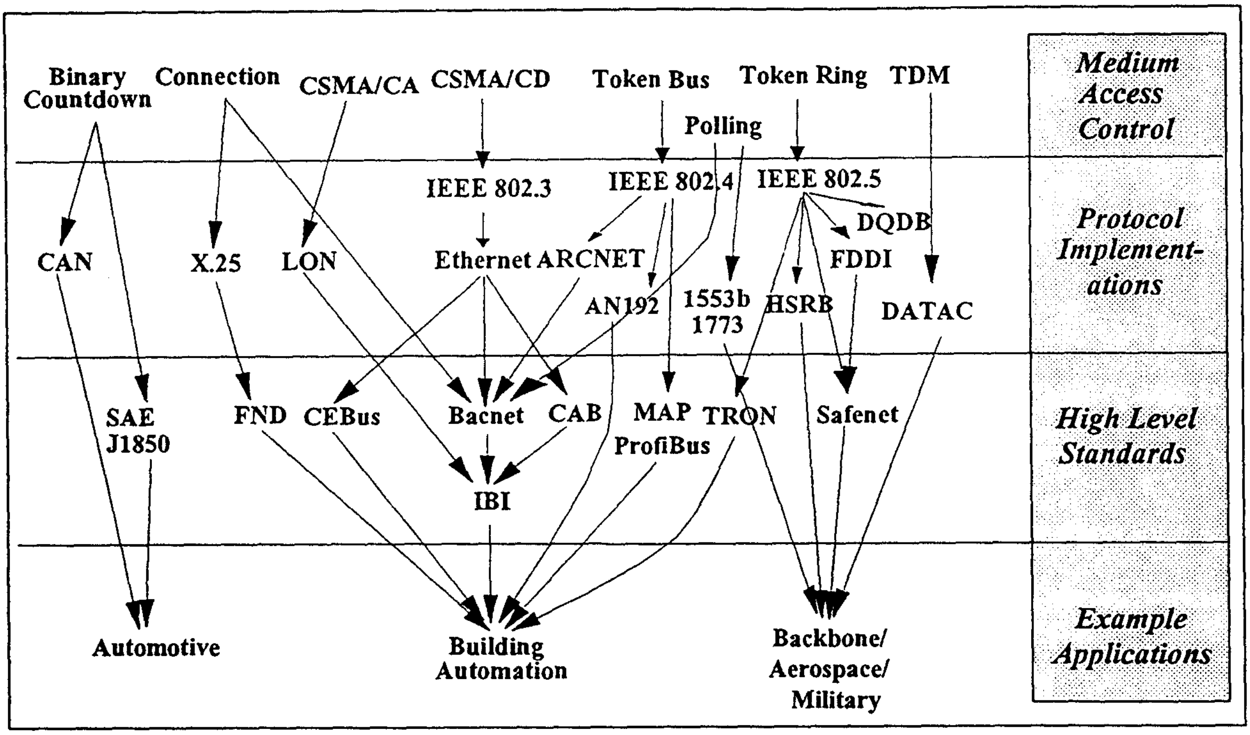


**Figure 1: "Standard" Protocol Family Tree**

<u>Time Division Multiple Access (TDMA)</u>

TDMA is heavily used in satellite communications[7], but is applicable to local area networks as well. In this protocol, each node transmits during its uniquely owned preallocated time slot. To maintain clock synchronization among all the nodes, a bus master broadcasts a frame sync signal before each round of messages. Like polling, TDMA is a simple protocol with deterministic response time that is well suited for balanced (evenly distributed), fixed length messages. Weaknesses include the bus master constituting a single-point-of-failure and bandwidth wasted by slots reserved for idle nodes. If a slot is not being used in some variations of TDMA, all stations can advance to the next slot early to conserve bandwidth (variable length TDMA). Time based protocols have been popular in military aviation applications. For example, **DATAC**[8], Digital Autonomous Terminal Access Communications, is being used by NASA and Boeing.

<u>Token Ring</u>

In a token ring, the nodes are connected in a ring-like structure using point-to-point links. A single token signal (special string of bits) is passed from one node to another around the ring.

The holder of the token has access to the network. This protocol offers a bounded latency, high throughput during heavy traffic, and global prioritization. Under light traffic, token ring has moderate token passing overhead. Initialization of the token message, detection of dual tokens or token loss adds to the complexity of the protocol. Many users are concerned that a break in the cable or a failed node can disable the network. However, node bypass hardware and dual rings can be used to address this concern[9]. Since the ring topology is unidirectional, it is well suited for fiber optics. Consequently, many LANs and Wide Area Networks (WANs) are moving to this type of protocol. For example, **FDDI** (Fiber Distributed Data Interface) uses dual counter-rotating rings to achieve higher reliability than bus or star topologies[10].

Token Bus

The operation of token bus is very similar to the token ring — a token is passed over the bus from one node to another creating a virtual ring. It works well under heavy traffic with a high degree of predictability. However, global prioritization of messages is inefficient, and latency under light loads is higher than for token ring because sharing a single bus precludes concurrent communication among logically adjacent nodes. Unlike unidirectional token ring, a break in the cable or a failed node does not disable the entire network. A lengthy reconfiguration process, where each node identifies its neighbors, is used to maintain the virtual ring when nodes are added or deleted from the network. Because bus-like topologies are well suited for manufacturing plants, **MAP**[11], Manufacturing Automation Protocol, adopted this protocol. Additionally, **ARCnet**[12], Attached Resource Computer Network, uses this protocol for LAN connectivity and process control. Adaptive Networks' **PLC-192** power line carrier chip uses a hybrid token bus protocol: under light traffic, nodes dynamically join and leave from the logical ring; under heavy traffic, all nodes join the ring to maintain stability[13].

Binary Countdown

In binary countdown, also known as the Bit Dominance Algorithm, all nodes wait for an idle channel before transmitting. Competing nodes resolve contention by broadcasting a signal based on their unique node identification value. During this transmission, a node drops out of the competition if it detects a dominant signal opposite to its own; thus, if a "1" signal is dominant, the highest numbered transmitting node will win the competition and gains ownership of the channel (Figure 2). It is also possible to arbitrate using unique message ID values rather than the node IDs to implement globally prioritized messages. This protocol has good throughput under light loads. A problem is that since all messages are prioritized, it is difficult to achieve fair access for all nodes under heavily loaded conditions. Using this protocol, Bosch developed a complete Controller Area Network (**CAN**[14]) specification for the automotive applications. The Society of Automotive Engineers standard **SAE J-1850**[15] also uses this protocol.

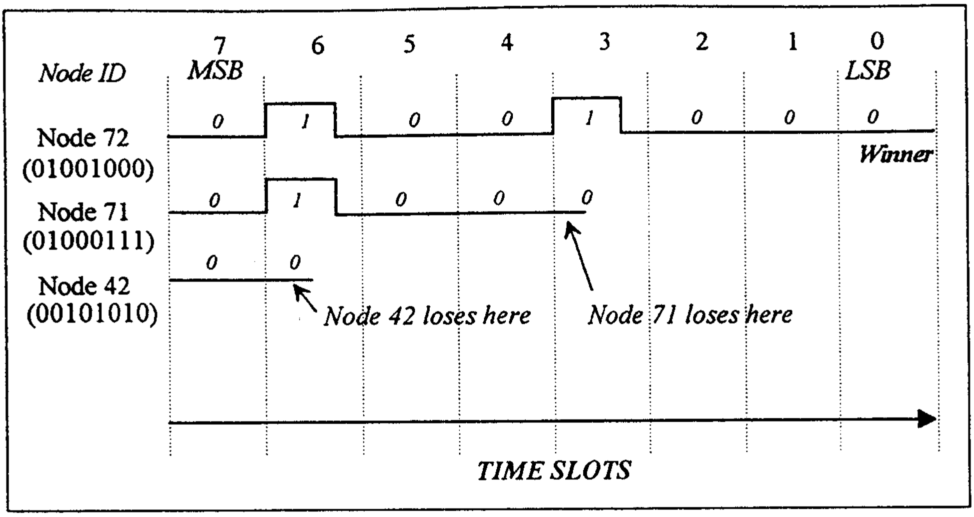


**Figure 2: Bit Dominance Procedure**

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

CSMA/CD has been widely researched with a large number of published variations[9,16]. In the simplest case, a node waits for the bus to go idle before transmitting. If multiple stations transmit simultaneously (within two bus propagation delays), the messages collide. The nodes must detect this collision, and resolve it by waiting a randomly generated time before retrying. The key advantage to this protocol is that it supports many nodes and allows the processors to enter and leave the network without requiring network initialization and configuration. Thus, for light traffic, MAC overhead is very small. However, under heavy traffic the MAC overhead is unbounded, leading to a protocol with poor determinacy. Furthermore, detecting collisions requires additional analog circuitry which translates to higher costs and difficult implementation in many embedded systems. The popular **Ethernet** protocol is based on CSMA/CD.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

Many researchers have developed hybrid protocols that combine the light traffic efficiency of CSMA/CD with heavy traffic efficiency of token-based protocols. The resulting protocols are often called CSMA/CA, or collision avoidance algorithms.

As in CSMA/CD, nodes transmit after detecting an idle channel. However, if two or more stations collide, a *jam* signal is sent on the network to notify all nodes of collision, synchronize clocks, and start contention slots. These time slots, typically just over two propagation delays long, are assigned to each of the stations. If the number of slots equals the number of stations, the protocol is called Reservation CSMA or RCSMA. The RCSMA variation works efficiently under all traffic conditions, and global priorities can be assigned through slot allocation[17]. Using rotating slots (slots that change positions after each transmission), fairness can be maintained and latency can be predicted. However, because of the one-to-one relation of the node to the slot, RCSMA is not practical for a network with a large number of nodes. Echelon's LON[18] (Local Operating Network) avoids this constraint by dynamically varying the number of slots (in some cases, fewer slots than stations) based on expected traffic and handling the case when multiple transmitters attempt to use the same slot.

## A HAND-WAVING COMPARISON

In the above discussions we have summarized the major MAC protocols and noted clear differences. Figure 3 shows some of the common traits and the relationships between various MAC protocols.

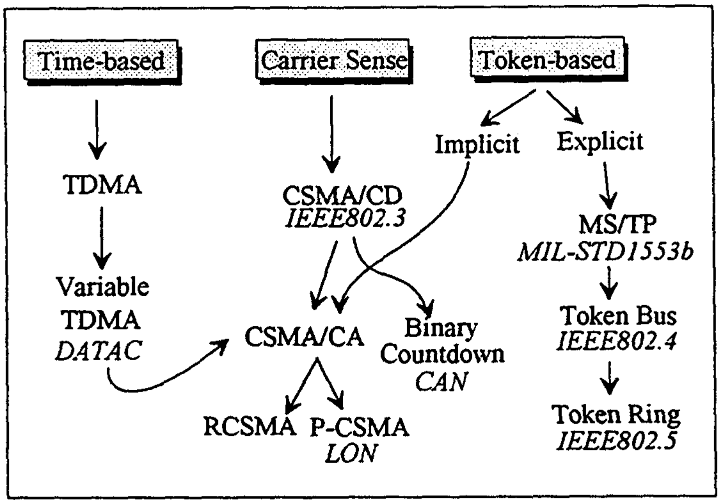


**Figure 3: Media Access Relationships**

Our opinions on the characteristics of all the media access protocols are compiled in Table 1. The important points to notice are:

Polling, TDMA, and connection-based protocols are simple, but do not provide sufficient flexibility for advanced systems.

Token-based protocols are predictable, but require complex software to maintain robustness.

CSMA/CD is a poor protocol for hard real-time systems with heavy traffic.

The collision avoidance protocols provide the best balance for embedded system requirements.

## THE WORLD OF STANDARDS

With the understanding of the MAC protocols and sample implementation standards, we can discuss the high level standards. Most of these standards lack specific hardware to go along with the published specifications. These standards have been developed by many public organizations and corporate alliances at industry, national, and international levels. As one would expect, progress is slow and consensus is minimal. To achieve consensus, some

| 👍=Good 👉=Ok 👎=Poor | Efficiency Light Traffic | Efficiency Heavy Traffic | Real-Time | Prior-itization | Robust-ness | Physical Layer | Cost/Node |
|---|---|---|---|---|---|---|---|
| Connection | - | - | 👍 | - | 👍 | 👉 | 👉 |
| Polling | 👎 | 👉 | 👍 | 👎 | 👎 | 👍 | 👉 |
| TDMA | 👎 | 👍 | 👍 | 👎 | 👎 | 👍 | 👉 |
| Binary Cnt. | 👍 | 👍 | 👉 | 👍 | 👍 | 👉 | 👍 |
| Token Bus | 👉 | 👍 | 👍 | 👉 | 👉 | 👍 | 👍 |
| Token Ring | 👍 | 👍 | 👍 | 👍 | 👉 | 👉 | 👍 |
| CSMA/CD | 👍 | 👎 | 👎 | 👉 | 👍 | 👍 | 👍 |
| CSMA/CA | 👍 | 👍 | 👍 | 👍 | 👍 | 👍 | 👍 |

**Table 1: Media Access Tradeoffs (a hand-waving approach)**

organizations are compromising by endorsing multiple protocols sponsored by members, resulting in standards and metastandards.

While high-level standards may ultimately yield benefits for high-level application interoperability, the compromises involved in permitting multiple physical implementations within a standard umbrella will likely impede standardization and cost reduction of actual communication hardware. For example, in the building automation industry, the Intelligent Building Institute (IBI) standard encompasses LON, CAB (Canadian Automated Building), and BACnet (Building Automation and Control Network). The MAC level of BACnet, in turn, allows the use of ARCnet (Token Bus), Ethernet (CSMA/CD), MS/TP, or a dial-up (connection oriented protocol). So, a product that conforms to the IBI standard could in fact use CSMA/CA, connection-oriented, polling, CSMA/CD, or token bus protocols at the hardware level.

In Japan, a consortium known as TRON[19] (The Real-Time Operating System Nucleus), is attempting to develop open standards for all information processing systems. They have defined the BTRON specification for business computing, CTRON for telecommunication industry, ITRON for industrial applications, and MTRON for Macro networking. In particular, the μBTRON[20], a specification based on the token ring, is aimed at networking real-time microprocessors in home, office, building, and automobile automation. However, TRON's development in the embedded arena is limited.

In Europe, several standards have been developed: Batibus in France, Profibus (MS/TP) and FND (X.25) in Germany. But their effect on the American market remains to be seen.

## ATTRACTIVE OPTIONS

There are many options that would be ideal for a particular application, and one should consider all reasonable options within design and business constraints. Nonetheless, we think that based on real-time performance, cost, and hardware availability, the following three protocols are attractive.

1. **ARCnet**: this token-bus-based protocol has transceiver chips (COM20020), a PC-based development system, and various communication peripherals (routers, repeaters, etc...) for a reasonable cost (from Standard Microsystem Corporation). In fact, SMC's new COM20051 chip integrates the COM20020 with a 8051 microcontroller.
2. **CAN:** this binary countdown based protocol is available from Intel (82527) and a Signetics/Phillips collaboration (8xC592). The 8xC592 combines the CAN protocol with the 8051. Development systems and supporting peripherals are offered by the chip vendors and other third-party consultants. The costs could drop significantly if automotive companies decide to endorse CAN based on studies they are currently performing.
3. **LON:** this CSMA/CA based protocol is a contender for the *de facto* standard in the control industry. LON interfaces are available for a variety of media and provide a large number of predefined network services in silicon. Chips are available from Motorola and Toshiba.

In selecting one of the standards in the family tree, we recommend that you give due consideration to the Media Access mechanism to avoid real-time performance problems later in the product life cycle. The MAC protocol should provide efficient use of available bandwidth, a flexible priority mechanism, bounded delays for messages, and robustness to failures.

Looking at the family tree, it is clear that a strong communication standard for embedded systems is not here yet. To the degree that differing applications have different requirements, a single hardware standard isn't possible. Furthermore, it appears that higher-level standards incorporate multiple protocols in response to political and business considerations rather than technical considerations. So, choices must be made based not only on capability, but also market share, and a prediction of the direction of standards in the future.

## REFERENCES

[1]Modiri, Nasser, "The ISO Reference Model Entities," *IEEE Networks Magazine*, vol. 5, no.4, pp. 24-33, July 1991.

[2]ASHRAE 135P, *BACnet: A Data Communication Protocol for Building Automation and Control Networks*, August 1991.

[3]Jordan, Larry, *System Integration for the IBM PS/2 and PC*, Brady, New York, NY, pp. 271-282, 1990.

[4]MIL-STD-1553B, *Military Standard Digital Time Division Command/Response Multiplex Data Bus*, September 1986.

[5]MIL-STD-1773, *Military Standard Fiber Optics Mechanization of an Aircraft Internal Time Division Command/Response Multiplex Data Bus*, October 1989.

[6]Zebermann, C., Kaster, L., and Rake, H., "Profibus - Open On-site Communication," *Proceedings of the 11th Triennial World Congress of the International Federation of Automatic Control*, pp. 143-146, August 1990.

[7]Stallings, W. S., *Data and Computer Communications*, 3rd ed., Macmillan, New York, 1991.

[8]National Semiconductor Corporation, "ARINC 629 Communication Intergrated Circuit - DATAC Data Terminal IC," ver. 2.0, Military/Aerospace Products Group, March 1992.

[9]Tanenbaum, A. S., *Computer Networks*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1989.

[10]Parker, Karen and Kaufman, Steve, "Fiber-Optic Links Hold Key to Fast LANs of the Future," Electronic Design, vol. 37, no. 25, pp. 46-53, December 1989.

[11]Rizzardi, Victor A., Understanding MAP, Manufacturing Automation Protocol, First Edition, 1988.

[12]Hoswell, Katherine S. and Thomas, George M., *ARCnet Factory LAN premier*, Contemporary Control Systems Inc., Second Printing, Downers Grove, Illinois, 1988.

[13]Gershon, R., Propp, D., and Propp, M., "A Token Passing Network for Powerline Communications," *IEEE Transactions on Consumer Electronics*, vol. 37, iss. 2, pp. 129-134, May 1991.

[14]Bosch, *CAN Specification*, ver. 2.0, Robert Bosch GmbH, Stuttgart, 1991.

[15]SAE J1850, *Class B Data Communication Network Interface*, July 1990.

[16]Bertsekas, D., and Gallager, R., *Data Networks*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[17]Chen and Li, "Reservation CSMA/CD: A Multiple Access Protocol for LANs," *IEEE Journal on Selected Areas in Communications*, February 1989.

[18]"Enhanced Media Access Control with Echelon's LonTalk Protocol," *LonWorks Engineering Bulletin*, Echelon Corp., August 1991.

[19] *The TRON Project 1992*, Tron Association, June 1992.

[20]Tanaka, K., Yasuyki, S., Tamai, K., Tsunoda, S., and Kato, H., "Performance Evaluation of the μBTRON Bus," *Proceedings of the Ninth TRON project Symposium*, pp. 40-45, 1992.

# Proceedings

*of the Fifth Annual*

# Embedded Systems Conference

*Volume 2*

Sponsored by Miller Freeman, Inc. and *Embedded Systems Programming*

ISBN # 0-87930-306-9



Miller Freeman, Inc. • 600 Harrison Street • San Francisco, CA 94107 • (415) 905-2200