

12-18-02

60434018

APPROV

PTO/SB/16 (10-01)

Approved for use through 10/31/2002. OMB 0651-0032

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Express Mail Label No. **EU 147 535 111 45****INVENTOR(S)**

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
AXEL SCOTT	FUCHS ANDREWS	PARK RIDGE, IL LOS ALTOS, CA

☐ Additional inventors are being named on the _____ separately numbered sheets attached hereto**TITLE OF THE INVENTION (500 characters max)**A DISTRIBUTED COMMUNICATION AND COMPUTING
METHOD ACROSS MULTIPLE COMMUNICATION LINKS FOR EMBEDDED SYSTEMS

Direct all correspondence to:

CORRESPONDENCE ADDRESS☐ Customer Number

Type Customer Number here

Place Customer Number
Bar Code Label here

OR

☒ Firm or
Individual Name

COGENIA PARTNERS, LLC c/o Axel Fuchs

Address

233 N HAMLIN AVE

Address

City

PARK RIDGE

State

IL

ZIP

60068

Country

USA

Telephone

847-384-1641

Fax

240-248-8933

ENCLOSED APPLICATION PARTS (check all that apply)☒ Specification Number of Pages

14

☐ CD(s), Number☐ Drawing(s) Number of Sheets☐ Other (specify)☐ Application Data Sheet. See 37 CFR 1.76**METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT**☒ Applicant claims small entity status. See 37 CFR 1.27.☒ A check or money order is enclosed to cover the filing fees

#2531

☐ The Commissioner is hereby authorized to charge filing
fees or credit any overpayment to Deposit Account Number:☐ Payment by credit card. Form PTO-2038 is attached.FILING FEE
AMOUNT (\$)

80.00

The invention was made by an agency of the United States Government or under a contract with an agency of the
United States Government.☐ No.☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____

Respectfully submitted,

SIGNATURE

[Signature]

TYPED or PRINTED NAME

AXEL FUCHS

TELEPHONE

847-384-1641

Date

12-17-2002

REGISTRATION NO.

(if appropriate)

Docket Number:

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Provisionary Patent Disclosure:***“A Distributed Communication and Computing Method across Multiple Communication Links for Embedded Systems”*****Inventors:**

Axel Fuchs and Scott Andrews, Cogenia Partners LLC

Background of the invention – State of the art:

With the decreasing cost of computing and communications, more and more industrial and automotive systems are controlled by one or more microcomputers. In an average car one can find up to 70 micro controllers (computing nodes), see figure 1. With additional cost reduction in communication and computing, this amount is likely to grow to more than 100 computing nodes per vehicle. In addition, telematics communications link the vehicle to a larger communication infrastructure outside of the embedded system in the vehicle.

With this increasing amount of network connectivity, the process of inter-node information sharing is complex and in some networks it is impossible to relay information between two distant nodes. Today's embedded systems are not designed to easily network with each other and they control complex systems with discrete digital and analog input / output (I/O). In most cases the applications on distributed systems exchange messages through only one physical network using a very simple application level messaging protocol. While today's functions are mainly independent of each other, the need for inter-connection between nodes is increasing in the future. For instance, a hybrid-electric vehicle will need a very tight link between the engine controller and the brake controller.

For embedded systems, each industry has defined a variety of control bus systems to link individual computing nodes together. The automotive industry has developed the Controller Area Network (CAN) in the 1980s and subsequently added new communication bus systems such as MOST, IDB, LIN, and FlexRay.

The industrial automation industry has followed a similar route with the IEE488 bus, the Field bus, and the Lon network.

Each of those communication bus systems create a singular communication and computing environment that cannot bridge into other networks below the application program level.

Echelon has designed a network using the Lon communication standard across multiple physical layers (US patent No. 5,513,324: Method and apparatus using

Axel Fuchs and Scott Andrews 12/16/2002

network variables in a multi-node network), but does not allow for multiple data-link layers as used in the automotive industry. Echelon has also introduced the concept of network variables in such networks that can be made available to distant nodes without direct physical links (US Patent No 5,737,529: Networked variables). And United States Patent 6,185,466 (Distributed digital control system including modules with multiple stored databases and selector) introduces the concept of databases in computing nodes, but does not place those databases in communication gateways for inter-networking functionality.

To implement this inter-network communication, the industry has defined several types of gateways or bridges. This approach involves translation of the various network protocol layers from one form to another, see figure 2. Following the classical approach of digital communications, in each step towards abstraction from the physical transportation layer, the protocol stack adds an additional header to the message sent through the network. While it is effective to abstract the application from the network complexity, this approach has several distinct disadvantages:

First, the gateway must perform a variety of computations and transformations on every message. As a result, the number of transformations becomes large as the number of interconnected networks grows. For N different networks, the number of transformations processed by the gateway increases as $N(N-1)$.

Second, with each layer of abstraction the message length grows and therefore decreases the throughput of the system or adds latency to real-time control loops.

Third, in many situations there is a one-to-many relationship between devices that generate data values and devices that use those values. In a system with multiple such user devices operating on multiple heterogeneous networks, the gateway must transform the messages containing the same basic data multiple times between each of the networks being served.

Fourth, in real-time embedded systems, the messages between distant nodes in the network need to comply with requirements for latency and jitter. Although one can define time-slots for each value on a specific communication bus, with the standard networking scheme it is impossible to forward the time sensitive information to a different physical communication link without introducing additional unpredictable latency and jitter through the required protocol transformations.

Fifth, current networking concepts can only cash information in pipes that are associated with the protocol stack. Without storage concept in a gateway that allows random access from multiple communication networks, it is impossible to exchange information between different predictable timeslots in each connected network.

Besides the use of optimized proprietary protocols, there are also proposals to use the standard Internet protocols for the embedded space. Even though the Internet and its communication protocols have shown the power of connecting random computers through heterogeneous networks, its paradigm can not easily be transferred to the embedded computing domain, because:

- 1) Distributed Internet applications exchange asynchronous messages that do not have guaranteed response times as required by real-time embedded applications
- 2) Embedded systems need fail safe redundancy and protection against attacks. It can be shown that the Internet is relatively fail-safe as long as enough connections exist. Some Web-Sites are available at mirror-site locations to improve access to information. But the Internet is not able to avoid dedicated attacks as long as intruders can execute application programs on remote computers.
- 3) The Internet is built on relatively "heavyweight" communication protocols that are not very well suited for the "low-latency" and "low-resource" embedded environment.

Overview of the invention

This invention addresses the shortcomings of traditional computer networks with six distinct innovations:

- 1) The concept of multi-mode storage to communication gateways links two or more communication networks via a bulletin board (see figure 3). The bulletin board is a multi-mode storage that can be thought of as an extension to shared memory that can be accessed by any process in a node of the attached networks. There may be multiple hierarchical layers of bulletin boards depending on the topology of the communication system. A shared memory represents a local bulletin board that only processes on a local processor node can access. The bulletin board increases the network efficiency by reducing the number of transactions needed to access remote variables.
- 2) The concept of a direct-access bulletin board that does not require a network layer on each node of the network. Even though this restricts the reach of each node to only the next gateway, this still allows cross-network variable sharing with very efficient data-link communications.
- 3) The concept of hierarchical bulletin board management that allows restriction of information access to certain levels in a network, but still allows the replication of information to other nodes in the network. This paradigm follows the path of reducing the information amount from the leaves of the network to central control and diagnosis hubs.
- 4) The concept that a gateway can host an assembly of bulletin boards or embedded database that allows operations on bulletin boards and allows storing the result in yet another bulletin board as part of the same

database. This extension allows definition of a set of data processing operations that would be done once in a network and would be instantly available for connected nodes. Examples for operations are sensor data filtering, state observers, diagnostics, integrity checks, fail-safe controllers, etc.

- 5) The concept that a embedded communication and computing network can run in multiple modes in order to provide for a guaranteed deterministic behavior of the system. This property can be achieved by only allowing change to the configuration and/or the functions (SW code) in a secured configuration and upgrade mode. If the network is booted in the normal operating mode, all processors execute the existing code and only allow data sharing through the bulletin boards. The emergency or debug mode lets the network run in a fail-safe reduced operation mode or in a diagnostic mode that allows inspection of the system, while it is running. For each operating mode, the gateway can store a processing image on the bulletin board. The advantage of this procedure is that only the communication hubs need to deal with secure data transfer and encryption while the peripheral nodes in the network can be relative simple in design.
- 6) The concept of designing the topology of a distributed computing and communication system independent of the definition of the individual functions that the network performs.

Detailed Description of the presently preferred embodiments of the invention

The current invention is a means for creating an information interconnection between two or more heterogeneous communication networks. The approach uses a common, or shared storage system that is connected to all of the system networks through network interfaces. A critically important feature of the bulletin board approach is that the complexity of the bulletin board grows linearly with the number of networks (as opposed to as $N(N-1)$ for the gateway approach), and in one-to-many situations the number of message transformations is half that of the gateway approach.

Each network interface translates between the network protocols for its associated network, and writes its variables into defined memory locations in the shared storage device. The shared storage device is called a Bulletin Board. The bulletin board may operate in one of two modes.

Mode 1 is used for reading parameters of sensors on the networks. This read-only mode is implemented by having each sensor device, or device capable of generating a value for a defined variable, write that value into the bulletin board

on a regular periodic basis. The update period is determined as part of the overall system design.

Mode 2 is used when a device on one network seeks to command a process (device) on another network. In this case the controlling device writes the control parameter into the bulletin board. The controlled device is instructed to poll the bulletin board on a regular basis. The period of this polling is determined by the specific design of the system, and the necessary response times for the control action. Alternatively, the bulletin board can be configured with a triggering scheme whereby a change in a memory location associated with a control parameter will initiate a request to the controlled device and send the parametric control information. This second approach is slightly more complex, but may be advantageous in high speed control situations.

Figure 1: Typical Automotive Network

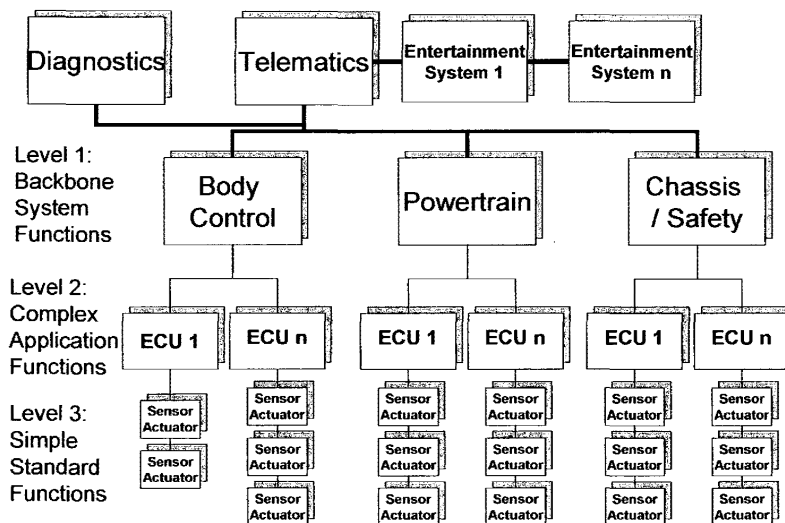


Figure 1 shows a typical automotive network. The various Electronic Control Units (ECUs) control complex applications such as engine control, brake control, or diagnostics. They are either connected to sensors and actuators via discrete links or simple standard functions such as sensors and actuators are organized into separate sub networks.

These complex functions such as braking, engine-control, etc. are then grouped into the backbone system functions for the car, such as body control, power train and chassis. The backbone also includes the vehicle's high level functions such as diagnostics, telematics and entertainment systems.

Figure 2: Classical SW Layers for communication network middleware

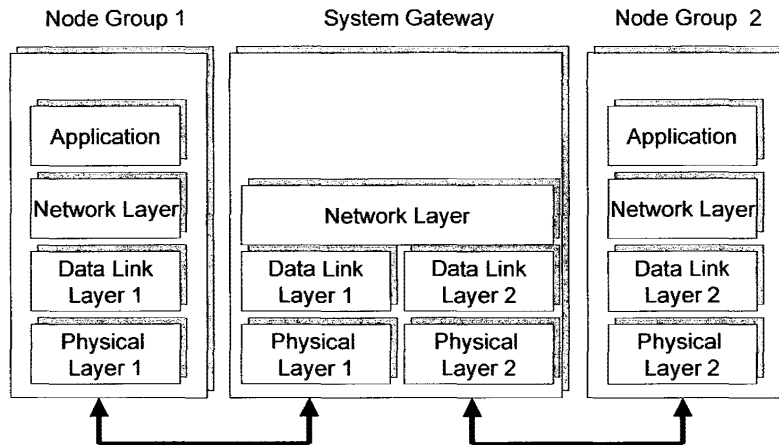


Figure 2 illustrates the logical architecture of the interconnection between two heterogeneous networks. The connection is fundamentally implemented at the physical interface (the wire, fiber or electromagnetic wireless interface). Each of the higher level layers (data link, network, etc) deals with specific features of the communication process. In practice these layers are represented in a message by "header" bits that contain information about that layer of the network being used to send the message. This is the well-known OSI data communication process. Using this model, each communicated message must be processed at each layer to remove (and use) the associated header information for that level. Once all layers are processed the remaining bits represent the datum or core information carried by the overall message. To communicate between two heterogeneous networks, this process must then be repeated in reverse by adding back the header information for the various layers of the second network, and eventually putting the message onto the second network's physical link.

Figure 3 and 4 illustrate the concept of a direct access Bulletin Board, a communication gateway with network storage. In the embodiment of the bulletin board access manager described in figure 3, any process can access network storage via a single network interface. This approach still requires a network layer in each processing node and therefore adds overhead to the communications. The system described in figure 4 has a multi-network access interface (essentially a processing capability that can interpret and apply the header information for a variety of networks) and the bulletin board read/write memory. Now, the individual processing nodes do not need to know about the existence of multiple networks. Each variable can be accessed from all

connected physical networks in their proprietary format. Thus the normalization of the information has only to be handled at the gateway

Figure 3: Network Communication Concept with bulletin board

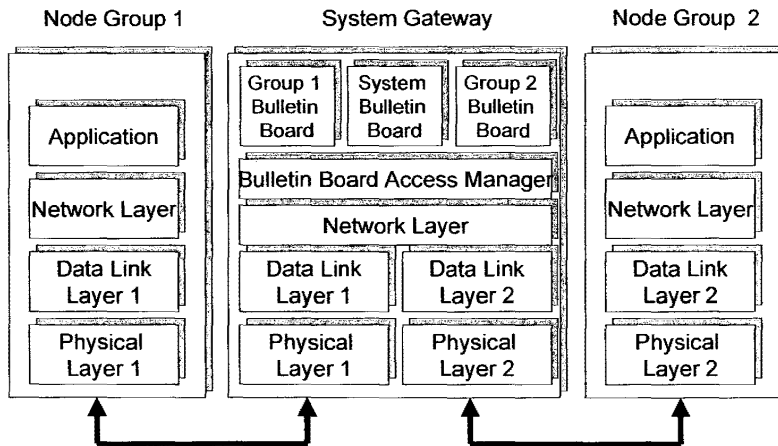
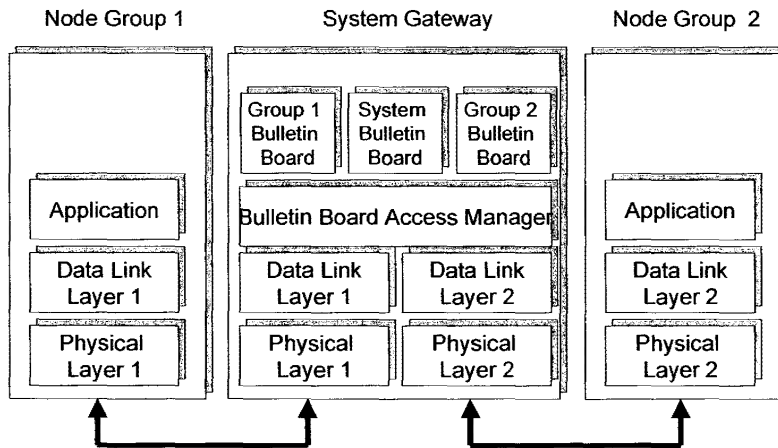
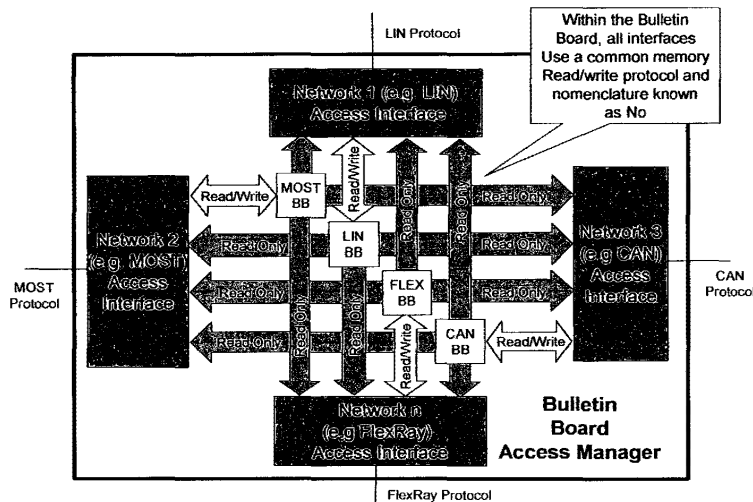


Figure 4: -Direct Access Bulletin Board



The bulletin board access manager is described in more detail in figure 5. The memory of the bulletin board is subdivided into areas that nodes on each network can read from and write into and other areas that a network may only read from. Thus the bulletin board manager provides mechanisms for access control.

Figure 5: Architecture of Direct Access Bulletin Board



This may be extended to the concept that the writable memory areas are unique for each network, so only one network may write information into the memory while the bulletin board access manager copies the information in other memory areas that are accessible from any other authorized network in a standard format.

The bulletin board may be further subdivided into memory locations associated with specific system variables. These are the parameters that are available within the system in question. In practice, these are the data contents of the various messages sent within the various networks. It is important to note that this approach requires every communication, command and data parameter to be treated as a system variable. While all systems operate in this way, few systems are designed from this perspective, because there has been no need to treat commands and messages as variables. This concept is shown in figure 6.

Figure 7 illustrates the concept of various hierarchies of memory management. In practice it is more efficient to allow each sub network and subsystem to place system variable data into local bulletin boards. This is because many system variables are primarily used only within their subsystem or sub network. By placing local information in a shared memory (local bulletin board), it can be used

by multiple processes on this processor node. A group bulletin board allows devices on a sub-network to share information with a minimum of network traffic. A system bulletin board allows access to system-wide variables and information.

Figure 6: Message conversion and protection mechanism

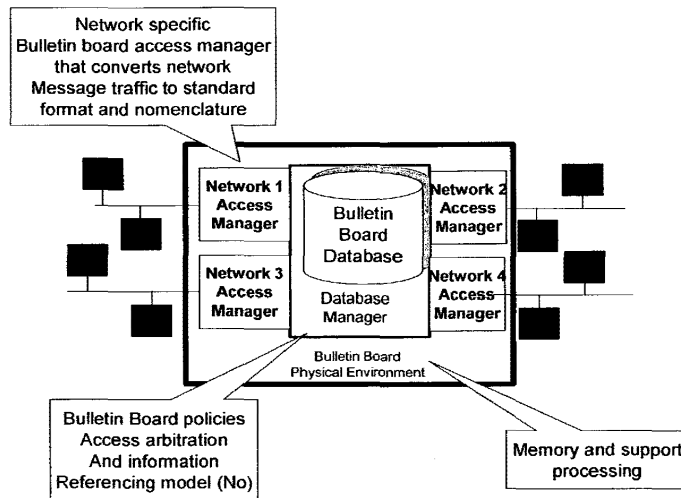


Fig 7: Hierarchical Bulletin Board Memory Management

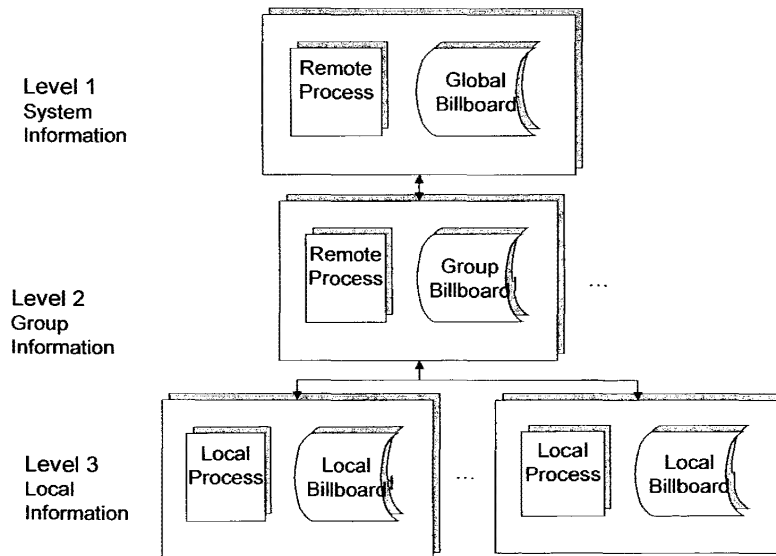


Figure 8 shows the concept of an extended bulletin board or an embedded real-time database. In this embodiment the communication gateway hosts a database, which hosts multiple bulletin boards with relational links between the variables in the bulletin boards. The relations are defined by data processing functions that the gateway can operate on bulletin boards to obtain new information that can be stored in yet another bulletin board.

Figure 8: Embedded Databases with enhanced bulletin boards

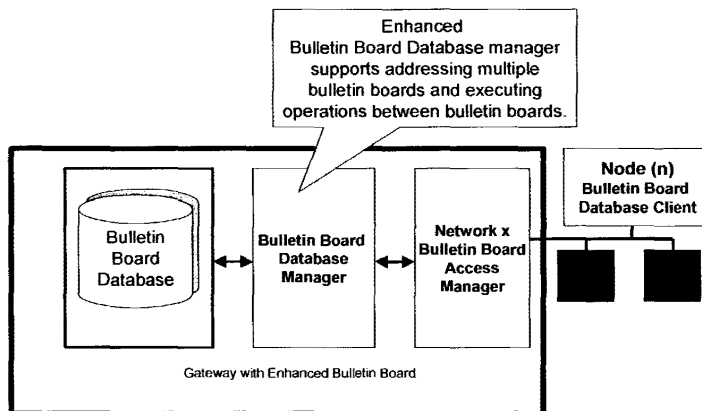
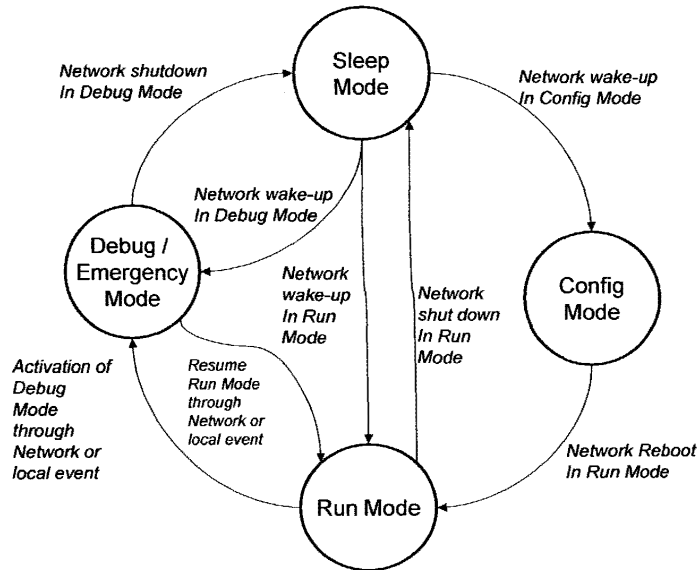


Figure 9 describes the various modes the communication network can be operated in. When the distributed computing and communication system wakes up from the sleep mode it can either enter a configuration and upgrade mode, an emergency or debug mode, or the normal operating mode. The root node in a bulletin board communication and computing system defines the mode based on the existence of external events, such as an external control command, a system failure, or failed integrity check. In the configuration mode, the system software and the information-sharing configuration can be updated via a secure communication link with encrypted commands. In the normal operating mode, the network configuration and the information sharing mechanisms can't be changed. That protection allows defining deterministic timing on all communication links. In the emergency mode, each processor installs and executes an alternate image that it downloads from the bulletin board, while it backs up the normal operating mode image on the bulletin board. This procedure allows the distributed system to reconfigure the communication and computing infrastructure based on available system resources. It also lets the system check on the integrity of sensors and actuators. In the maintenance and upgrade mode, an external

system can upgrade executable code images and the bulletin-board configuration via secure communication links.

Figure 9: Network state diagram



We Claim:

A distributed computing and communication network for embedded systems:

1. That spans over multiple physical communication links with diverse data link layers
2. That abstracts the networking function away from the functions that each application on a node provides. Thus the function of each network node can be designed without having detailed knowledge of the network and the type of network connectivity.
3. That exchanges information through "virtually shared memory" or bulletin boards
4. That provides multiple hierarchies of bulletin boards for applications in form of hierarchical bulletin boards (group bulletin board, global bulletin board).
5. That allows for data replication mechanisms between multiple bulletin boards.
6. That for each bulletin-board variable the user can define real-time constraints (time bound, not time bound, bandwidth, latency, jitter, time triggered or event triggered) and memory constraints (size, format). Each datum is defined by name, ID, size, type, and the data field

7. That allows the definition of processes that perform calculations or data processing on the data in one bulletin board and store the result of the operation in yet another bulletin board.
8. That can be operated in at least three distinct operation modes
 - a. Configuration and Upgrade Mode
 - b. Normal Run Mode
 - c. Emergency Run mode or Debug Mode
9. That only allows upgrades of functionality in a secure configuration mode
10. That allows the protection of changes in the configuration via network link security and command encryption.
11. That allows exchanging group variables across a local communication network without sending the full networking header between nodes on one physical network.
12. That allows for bulletin boards to update applications with a different information update rate as the bulletin board receives from the source.
13. That provides distributed event triggers independent of the link connection between the nodes in the network.
14. That provides failure redundancy across multiple independent links across diverse physical connections.
15. That provides access to group and global bulletin boards with guaranteed access times (latency), jitter, and bandwidth (throughput).

The design methodology of a "bulletin board" based communication system for embedded systems :

- a) That separates the network topology design from the application program design by defining the function first and then allocating the function to processor nodes.
- b) That places operations on network variables with its results on the processor nodes that host the bulletin board
- c) That automatically determines the number of links needed between the nodes following a power distribution of the amount of links per node analog to a scale-free optimal network.
- d) That the system calculates the optimal placement of billboards based on the performance requirements and the most efficient network topology.

Reference Patents:**United States Patent 5,513,324**

Dolin, Jr., et al. April 30, 1996

Method and apparatus using network variables in a multi-node network**Abstract**

An improved apparatus and method for communicating information in a networked system wherein network variables are employed to accomplish such communication. Network variables allow for standardized communication of data between nodes in a network. A first node may be programmed, for example, to sense certain information and to report the information as a network variable X. A second node may be programmed to receive the variable X and to control devices based on the current value of the variable. The present invention provides for defining connections between the first and second node to facilitate such communication and for determining addressing information to allow for addressing of messages, including updates to the value of the variable X, between the nodes.

Inventors: Dolin, Jr.; Robert A. (Menlo Park, CA); Einkauf; Robert L. (Fremont, CA); Kagan; Richard S. (London, GB2); Riley; Glen M. (Los Gatos, CA); Von De Bur; James M. (San Jose, CA)

Assignee: Echelon Systems Corporation (Palo Alto, CA)

Appl. No.: 207229

Filed: March 7, 1994

United States Patent 6,185,466

Nicewonger February 6, 2001

Distributed digital control system including modules with multiple stored databases and selector**Abstract**

A distributed digital control system includes a plurality of control modules which are interconnected by a communication link. Each module includes a processor unit for running a control program, and a plurality of system databases which correspond to nodes in the system. Each module further includes a selector such as a Dual Inline Package (DIP) switch for inputting a node number. The program automatically selects the database corresponding to the input node number when the module is powered up. The system preferably utilizes the industry standard LonTalk.RTM. protocol, with each database including Standard Network Variable Types (SNVTs) and Standard Configuration Parameter Types (SCPTs) for the respective node. A default database is also stored in each module, which is automatically used by the program if the system databases are not found.

Inventors: Nicewonger; Mark R. (Aptos, CA)
Assignee: Proteus Industries, Inc. (Mountain View, CA)
Appl. No.: 944745
Filed: October 6, 1997

United States Patent 6,128,315

Takeuchi October 3, 2000

Distributed control network system

Abstract

In the distributed control network system, the local control server has a just-in-time compiler with which the user transforms the control for each node described in Java into a code that can be executed on the node at the time of execution. Additionally, the local control server has a functional feature of transforming the protocol so as to make the nodes in its network able to communicate with the nodes in another network having a different protocol. There is also provided means for programming the control to be performed by each node and the flow of processing operations among the nodes in a GUI environment.

Inventors: Takeuchi; Yoichiro (Urawa, JP)
Assignee: Kabushiki Kaisha Toshiba (Kawasaki, JP)
Appl. No.: 990538
Filed: December 15, 1997