# Application of Embedded Systems In Automotives

**Pulla Reddy.M[1], Javeed Ali.S[2], Chanikya.K[3], Sai Manideep.K[4]**

1.Final Yr B.Tech student of Mechanical Engineering, Gates Engg. College, Gooty, Ananthapuramu, AP, IN: pr98091@gmail.com
2.Final Yr B.Tech student of Electronics and Communication Engineering, Gates Engg. College,Gooty: www.javed11@gmail.com
3.Final Yr B.Tech student from Department of Mechanical Engineering , Gates Engg. College,
Gooty: chanikya.k333@gmail.com 4.Final Yr
B.Tech student, Department of Mechanical Engineering, Gates Engg. College, Gooty: Mahesh.kumar167@gmail.com

## ABSTRACT

**The number of computer based functions embedded in vehicles has increased significantly in the past two decades. An in-vehicle embedded electronic architecture is a complex distributed system; the development of which is a cooperative work involving different manufacturers and suppliers. There are several key demands in the development process, such as safety requirements, real-time assessment, schedulability, composability, etc. Intensive research is being conducted to address these issues. This paper reviews recent technology advances in relevant aspects and covers a range of topics highlighted above.**

**Key Words:** In Vehicle embedded electronic architecture, ECUs, LIN, CAN

## 1. INTRODUCTION

An embedded system is typically a micro-computer system with one or few dedicated functions, usually with real-time computation constraints. Different from a general purpose personal computer, it is often embedded as part of a complete device. The usage of embedded systems is so widespread today, e.g. smart phones, programmable systems on chip (SoC), smart sensors, etc. These types of embedded systems include microprocessors, DSPs (digital signal processors), ASICs (application-specific integrated circuits), and FPGAs (field-programmable gate arrays).

It is worth mentioning that an embedded controller based on Artificial Intelligence is becoming more and more popular; intensive research in this domain has evolved. For example, proposes a lightweight method to implement the neuron-by-neuron process on embedded systems to correct the nonlinearities of many sensors and devices. present methods to implement a Fuzzy Logic Controller on reconfigurable FPGA systems. implements Gauss-Newton and particle swarm optimization algorithms to estimate sensor-node physical-position. The recent two decades has witnessed a trend in the automotive industry---a rapid growth in the percentage of cost of embedded electronic systems, more precisely the software components. As in 2006, the electronic embedded system constituted at least 25% of the total cost of a car and more than 35% for a high-end model. Top-line cars today may contain up to 100 ECUs (Electronic Control Unit). Each controls one or more of the electrical systems or subsystems in a motor vehicle networked over standard communication buses. Local area networks such as LIN, CAN, Flex Ray, Most and IDB-1394 are developed as such links. Considering the increase of complexity of embedded electronic architecture, the development of it has to integrate different hardware and software units provided by different vendors which raises the question vendors, which raises the question of “composability”.

This paper will review recent advances in the relevant technologies in the subsequent sections, including temporal isolation of software components, probabilistic approach of latency computation, and upgrading from single core ECUs to multicore ECUs.

## II. BASICS

CAN is a serial fieldbus communication network. CAN uses a “bus” to transmit messages between nodes in the network. The bus arrangement reduces the number of connections between nodes. Each node has a single 2-way connection to the bus. Point-to-point connections, by contrast, require an individual link between every node. The bus arrangement requires “n” connections for “n” nodes while a point-to-point arrangement requires “(n(n-1))/2” connections for “n” nodes. The bus arrangement provides a significant reduction in cost and reduces propagation delays. The fact that it is a serial bus indicates that only a single message can reside on the network at a time. CAN is a distributed “real-time” network used in the field (“Fieldbus”) where “real-time” events occur. basic CAN is “event-oriented” (event-triggered), meaning that messages are generated in response to the generation of events on the network. Nodes send messages following an action (receives a data frame) or a request for information (receives a remote frame). However, due to the nature of CAN message arbitration where the outcome of and the time required to resolve each arbitration is completely dependent on the value of the message ids at the time of arbitration, the times required to send and receive messages cannot be characterized deterministically. This is considered to be insufficient for “real-time” applications where “hard” real-time constraints impose strict requirements on the ability of network nodes to intercommunicate. In these applications, it is desirable to be capable of triggering deliberate messages at precise instants, avoiding and superseding arbitration. Transmission and reception of messages is temporarily performed during a deterministic time slot instant. To facilitate the adherence to a “time-oriented” (time-triggered) approach, an upper layer called the TTCAN (Time-Triggered CAN) layer was developed. As more functionality of the automobile is transferred from mechanical to electronic control, network topology complexity, number of gateways between networks and the need for portability between automobile platforms have steadily increased. The standard high-speed CAN bit rate of 1 Mbit/second has been superseded by 5 Mbit/second rates on a single-channel link and 10 Mbit/second rates on twochannel links (with redundant traffic possible) using the “Flex Ray” specification. “Hard” real-time requirements mean that “time-triggered” events are necessary, such that TTCAN-like messaging is employed. In addition, the bit rate (bandwidth) of the system is variable depending on the bus load required. Additional channels (links) are used to vary the bit rate (throttling) or when multiple nodes must send data at the same time. A master FlexRay node provides a “global reference time” from which all other nodes on the network derive their message timing.

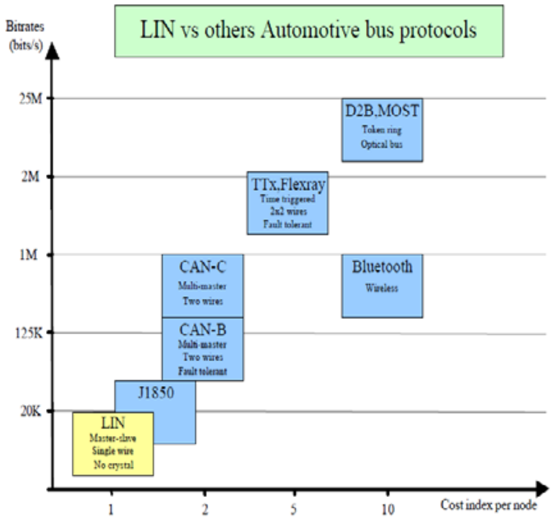


Fig.1LIN vs other Protocols

An optional “bus guardian” component manages Error Containment of messages, above and beyond the functionality provided by the CAN specification. LIN was developed as a simpler, more cost-effective alternative fieldbus technology for bit rates (up to 20 kbits per second) on par with Low Speed CAN. Like FlexRay, LIN has a single master LIN node which coordinates timing across the network of slave nodes. However, in LIN there is no message arbitration and all messages are considered to be deterministic with static latencies. LIN provides sufficient functionality at low cost (with a finite number of nodes). It lacks redundant slave message re-sending (only a master node controls routing) and is for low performance (bit rate, bandwidth) requirements. CAN is scalable and dependable due to its bus topology. However, CAN lacks deterministic scheduling for real-time events because message arbitration can in theory delay message routing indefinitely. LIN, TTCAN and FlexRay all provide deterministic scheduling with the aid of a master network node. FlexRay bandwidth is variable with multiple channels available. CAN, TTCAN and FlexRay support message re-transmissions following transmission of error messages.

## III. FRAMES

CAN delineates the types of “frames” (message formats) that can be sent over the bus into 4 main groups. “Data Frames” transport data between nodes. “Remote Frames” are requests for transmission of “Data Frames”.

“Error Frames” are messages indicating that an error has occurred (transmission bit error, reception bit error, etc). “Overload Frames” are used to request a delay of bus signal between the transmission of data or remote frames when a node is “overloaded” with data. Modern processors often do not require the use of the “Overloaded Frame”. TTCAN resides primarily in the OSI “session” layer. Synchronization of messages on the network is accomplished by a single master TTCAN node which assigns the remaining nodes on the network “Time Slots”. These time “windows” are the only times available for nodes to transmit. In general, each node has a different set of assigned windows in which it may transmit. Assignment is controlled by the master TTCAN node, and the number, duration and type of window assigned to each node will depend on the nature of the node and what the node is being used for in the network. Flex Ray “static segments” are used for synchronous real-time time-triggered events, and Flex Ray “dynamic segments” are used for asynchronous event-triggered events where one dynamic segment may demand more or less bus bandwidth than another dynamic segment. In this way, a single FlexRay frame can perform arbitration and no arbitration via a method similar to a single TTCAN Base Cycle . Master LIN nodes decide which messages may be transmitted on the bus and their timing. Slave LIN nodes are not allowed to send messages to the bus unless the master LIN node requests them to do so. Each message is called a “Frame” (similar to CAN) with a fixed format but a variable length. The main fields are the “Header”, “Response Space”, “Response” and “Interface Space”. The Header field is always provided by the master LIN node, and the Response field is provided by the transmitting node (either of the master or slave LIN nodes). CAN frames provide dependability and fault-tolerance via re-transmissions and error frames. Master LIN nodes control message frame re-transmission following a transmission error. Data correctness is ensured using CRC fields in LIN, CAN, TTCAN and FlexRay frames. “Global reference time” found in LIN, TTCAN and Flex Ray provides better schedulability of time-triggered messages, while trading off some efficiency and local node autonomy. FlexRay has greater extensibility to future platforms because of its variable bit rate throttling, support for fiber optics and inclusion of an optional bus guardian.

## IV.VEHICLE EMBEDDED ARCHITECTURE

CAN ECU/ node consist of a microcontroller, a CAN controller, a line driver and I/O This can be mimicked with E-blocks. This can be controlled with Flowcode.

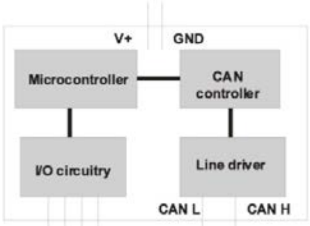


Fig.2 Data transmission

An Electronic Control Unit has the devices shown. In practice this can be implemented in a number of ways: for example you may find a microcontroller with an internal CAN controller and I/O circuitry and you may even find a single chip that has all these four blocks. The E-blocks solution is a bit blocky here, but they form a perfectly valid, fully working ECU.

Fig.3 E-blocks CAN controller and line driver

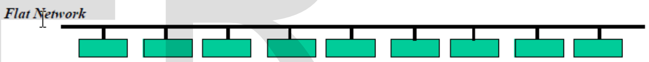


Fig.3 CAN based Network

## V. LIN (Local Interconnect Network)

Is a cost-effective and deterministic communication system for connecting ECUs with smart sensors, actuators and controls. Satisfy Need for a Standard for Sub Busses

Fig.4 CAN + LIN

## VI. LIN, CAN, and Flex Ray

Although many network protocols have been proposed over the years, the most significant are LIN (Local Interconnect Network) and CAN (Controller Area Network). First used in vehicles in 1994, CAN handled networking functions until LIN—more cost effective but lower performance—was introduced in 2002. The newest automobile network, Flex Ray, has only recently appeared in cars. LIN can be implemented at low cost and is perfectly suited for local control operations such as actuating door locks and electric window controls. LIN's low data rate (20 kbps) prohibits its use in more sophisticated and data-intensive applications such as motor control, braking, and suspension systems. That market is dominated by CAN, which comes in three flavors: Fault-tolerant CAN, high-speed CAN, and single-wire CAN, which is used primarily by General Motors. Cars are becoming more sophisticated every year, however, and network technology has to keep pace. So within the next year or two, Flex Ray will begin its significant market penetration. Flex Ray is a high-speed (10 Mbps) network well-suited for time-critical applications. It has been in development since 1999 and is now in use in some high end vehicles.

The figure 5 illustrates the market segmentation for in-vehicle networks over the next few years in terms of individual network connections, or nodes. Each node is the connection point on the network for an ECU (Electronic Control Unit) that supervises and controls some mechanical system.

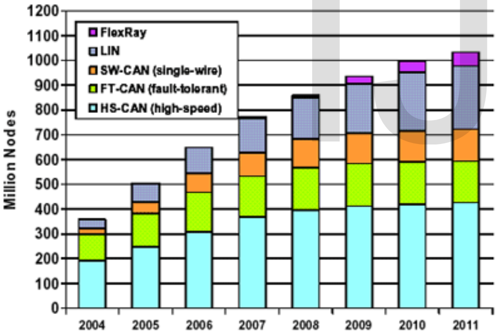


Fig.5 Usage of Varios protocols

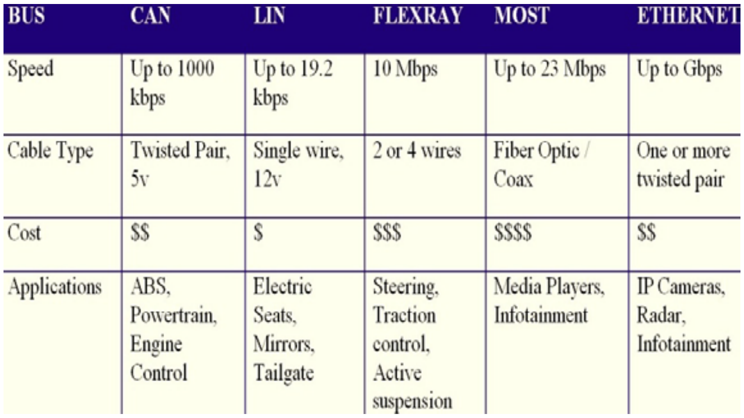
The Local Interconnect Network (LIN) protocol specification provides a low-cost, short-distance, and low-speed network, enabling the implementation of a new level of electronics intelligence in automotive subsystems. LIN operates under a CAN platform, but it doesn't require the robust data rate and bandwidth performance, or the higher cost, associated with CAN.

The typical automobile network is broken up into several sub networks. These include body control, power train, and multimedia networks. Depending on the speed and cost requirements, either CAN or LIN can be implemented. For example, a power train demands significant computational processing speeds. Yet, body electronics is more typically oriented to human-interface speeds, which lend themselves nicely to the 20-kbit/s LIN architecture. Window lifts, door locks, seat positioning, environmental controls, wipers, etc. are all ideal benefactors of a robust LIN implementation where wiring complexity and weight are dramatically reduced, and reliability is significantly enhanced. LIN is a single-master multi slave bus that communicates via a single wire, reducing wiring complexity as well as cost. Because this protocol is self-synchronizing, it allows the slave nodes to run from a low-cost RC oscillator. LIN and CAN don't compete with each other, but instead complement one another. On the one hand, CAN serves high-speed, error-sensitive needs and operates on a 5-V differential bus. LIN, however, serves low-speed, low-bandwidth requirements on a 12-V single-wire bus.

An interface is necessary between LIN's country roads and CAN's high-speed thoroughfares. A bridge node might consist of a microcontroller with an integrated LIN transceiver combined with a low-cost stand-alone CAN transceiver. This provides the intelligence that's needed to watch both buses and interchange data.

For instance, perhaps temperature information would be required by various systems within the automobile. For this application, indoor and outdoor temperature sensors may be incorporated into a car door's LIN network. The door's master node would place the data onto the CAN bus. From there, additional slave LIN devices in the rear-view mirror, or a heads-up display, might present the data to the driver, while the environmental control system could use the data to activate the air conditioner or heater and defroster grids.

Table 1Comparision of Various Protocols



| BUS | CAN | LIN | FLEXRAY | MOST | ETHERNET |
|---|---|---|---|---|---|
| Speed | Up to 1000 kbps | Up to 19.2 kbps | 10 Mbps | Up to 23 Mbps | Up to Gbps |
| Cable Type | Twisted Pair, 5v | Single wire, 12v | 2 or 4 wires | Fiber Optic / Coax | One or more twisted pair |
| Cost | $$ | $ | $$$ | $$$$ | $$ |
| Applications | ABS, Powertrain, Engine Control | Electric Seats, Mirrors, Tailgate | Steering, Traction control, Active suspension | Media Players, Infotainment | IP Cameras, Radar, Infotainment |

The CAN specification requirement for node-to-node oscillator tolerances of 1.5% and hardware-based error handling drive node costs up to well over $2.00. But, LIN's tolerances of 15% and the ability to implement its protocol entirely in software are moving node costs below $2.00.

## VII. ERROR MANAGEMENT

Detection of an error message on the bus occurs because CAN specifies that no message on the bus may consist of 5 or more bits which have the same polarity (i.e.: ..00000.., and ..11111.. are considered errors). It happens that the first 6 bits of an error message are all dominant bits (000000). Whenever a node identifies a 5-bit sequence of mono-polarity bits (or 6-bit sequence from an error message), it halts whatever it is doing and issues an error message. “Bit stuffing” is the process of inserting a reverse polarity bit immediately following the sequence of monopolarity bits. This “stuff bit” serves as a flag to receiving nodes that the prior mono-polarity bit sequence should not be an indication of an error state. Receiving nodes remove the identified “stuff bit” from the incoming message, and the remainder of the message is parsed. LIN is capable of performing error detection similar to CAN.

However, due to the “lightweight” nature of LIN, this functionality is severely reduced. Unlike CAN, LIN does not have error frames for signaling errors, but instead the slave LIN node that detects an error locally issues a diagnostic message, in the form of a standard data frame, to the master LIN node only. CAN error frames and LIN diagnostic frames provide dependability and fault-tolerance, as well as security against network failures, attacks, or electromagnetic interference. CAN “bit stuffing” provides scalability to larger CAN networks which may use an increased number of message ids.

## VIII. AUTOMOTIVE FIELDBUS SYSTEMS

Electronic vehicle system networks such as CAN, TTCAN, Flex Ray or LIN have many advantages. First, the cost of implementing the cabling for these networks is lower than traditional approaches. These are serial bus systems (all communicating components connected to the bus) which reduce the amount of cabling required over traditional point-to-point networks (all communicating components directly connected to all other communicating components) for the wiring harness of the vehicle and simplify vehicle assembly. In addition, the lower number of plug connections of a serial bus system corresponds to fewer wires and

Second, control is typically distributed to the individual Electronic Control Units (ECUs) locally positioned at the specific control system. Service technicians can plug into the network and access any unit on the network using a common software package, and be capable of communicating with any component in the network (manufactured by any Original Equipment Manufacturer, OEM or vendor). increases dependability and reliability.

Third, “hard real-time” requirements, parameters which must not deviate from strict limits (real-time braking control, real-time engine control) in order to avoid catastrophic events can be realized by meeting so called “real-time” operation levels.

This means that very fast response times to real-time physical events (less than 100 microseconds, which is unnoticeable to a human observer) are realized. This list of advantages supports the use of deeply embedded control systems in vehicles and is the rationale motivating the switch from mechanical systems to “brake-by-wire” and “drive-by-wire” electronic systems

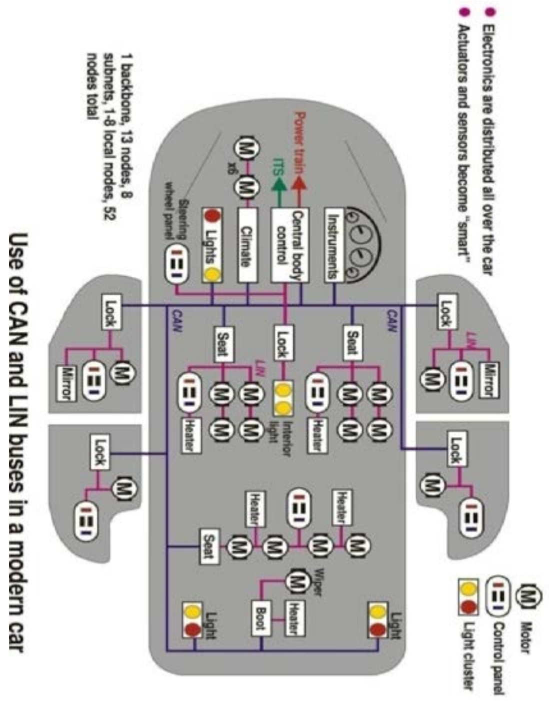
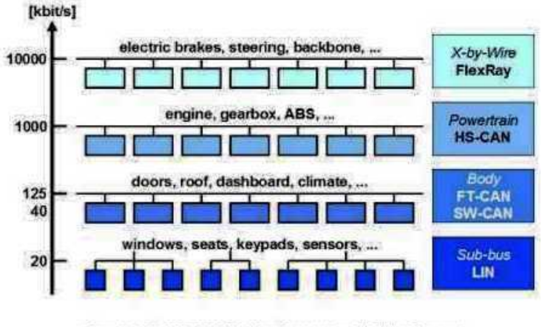


Fig.6 In-Vehicle Embedded system



Copyright © 1995-2008, Digi-Key Corporation. All Rights Reserved.

Fig.7Ranges of Data Tranmission

Accepting messages which have “message types” that they have been registered to recognize before-hand. This approach increases “network elasticity” by allowing new nodes to be added to the network without requiring prior registration with the other nodes, so long as these new nodes have been programmed to recognize the existing set of message types. Four commonly used electronic systems in passenger vehicles are CAN, TTCAN, FlexRay and LIN. In the following sections, we will discuss and compare each in detail.

## IX.CONCLUSION

In this paper, we have compared four important related features among different fieldbus networks. In particular, CAN is scalable and dependable due to its bus topology.

However, CAN lacks deterministic scheduling for real-time events because message arbitration can, in theory, delay message routing indefinitely. LIN, TTCAN and Flex Ray all provide deterministic scheduling. Flex Ray bandwidth is variable with multiple channels available. TTCAN, LIN and Flex Ray have a master node for handling scheduling of messages. CAN frames provide dependability and fault tolerance via re-transmissions and error frames. CAN error frames and LIN diagnostic frames provide dependability and fault-tolerance, as well as security against network failures, attacks, or electromagnetic interference. The two tables given below summarize the comparison. In particular, Table I, “Attributes of Automotive Networks”, shows several key factors useful for describing and comparing LIN, CAN, TTCAN and FlexRay networks. “Messaging” compares event triggered and time-triggered messages broadcast on the network. “Network Synchronization” compares the use of global reference time versus the use of message arbitration. “Node Control” shows how nodes in a network are controlled, via autonomous control or via a master/slave configuration. “Error Management” compares the strategies used by the networks to avoid, contain and recover from errors generated on the network. “Bandwidth” describes the various data rate capacities of the networks.

Embedded systems, especially in-vehicle embedded systems, are ubiquitously related to our everyday life. The development of embedded systems greatly facilitates the comfort of people’s life, changes our view of things, and has a significant impact on society. On the other hand, even though embedded systems technologies are becoming more and more mature, currently there challenges in technique and will be more with the ever growing speed and reliability demand from market. We expect more enlightening researches in this area.

## X.ACKNOWLEDGEMENT

We specially thankful to M.Sreekanth and Naveen kuamr both are working as Asst. Prof s at Dept of Electronics and Communication Engineering, Gates Institute of Technology, Gooty

## References

[1].Embedded systems by Rajkamal
[2] Mechatronics Electronics Control Systems in Mechanical and Electrical Engineering , W Bolton, 3/e, Pearson Edu. Press, 2005.
[3].Mechatronics, M.D.Singh, J.G.Joshi, PHI.
[4] D. Paret, Multiplexed Networks for Embedded Systems, Wiley, 2007
[5] D. Caro, Automation Network Selection, ISA, 2004, pages 93-106
[6] M. Dahleh, E. Frazzoli, A. Megretski, S. Mitter, A. Ozdaglar, P.Parrilo, D. Shah, "Network
[7] W. Voss, A Comprehensible Guide to Controller Area Network, Copperhill Technologies Corporation,

**PULLAREDDY MEKALA**

This was his first International Conference, an active member at IEEE and SAE. Experienced with 1 International Journal on PLC and presented at 4 National Level Symposiums

**JAVEED ALI**

This was his first International Conference and Journal publication, and presentations at 2 National Level Symposiums

**SAI MANIDEEP.K**

This was his first International Conference had 1 International Journal on PLC and presentations at 2 National Level Symposiums

**CHANIKYA.K**

This was his first International Conference presented at 3 National Level Symposiums