Survey on Automotive Network Communications Protocols

Rucha Rasane#1, Dr. Krupa Rasane#2 and Prof. Prabhakar Manage#3

#1 Student, Electronics and Communications, KLE Dr. M.S. Shesgiri College of Engineering and Technology, Belagavi, India

#2 PhD, Electronics and Communications, Jain College of Engineering, Belagavi, India

#3 Assistant Professor, Electronics and Communications, KLE Dr. M.S. Shesgiri College of Engineering and Technology, Belagavi, India

Abstract—This paper presents an overview of two automotive networked communications protocols LIN (Local Interconnect Network) and FLEXRAY developed by the FlexRay Consortium and are used to govern on-board automotive computing and communication between the ECUs (Electronic Control Units) and various other components of a car. In today’s highly automated world, with millions of cars crowding the streets and innumerable people asking for automatic features, in-vehicle communication and vehicle to vehicle communication have become a part of the modern evolution of the automotive industry. LIN and CAN protocols have already made their mark on the automotive industry. FLEXRAY however is yet to be fully exploited. As of today, FLEXRAY protocol has been accepted as the future of the automotive industry because of its advantages in performance and flexibility of design as compared to the other protocols. This paper peeks into the two protocols LIN and FLEXRAY and compares them with the CAN (Controller Area Network).

Index Terms—LIN, CAN, FLEXRAY, ECU, In-Vehicle Communication.

I. INTRODUCTION

Improving traffic safety has been a top concern for transportation agencies around the world. Cooperative vehicle systems which use sensors and wireless technologies to reduce traffic accidents can play a major role in making the world’s roads safer. Implementing these cooperative features is made possible by the use of various vehicle network communication protocols such as CAN, LIN, MOST (Media Oriented Systems Transport) and FLEXRAY. There are many bus systems used in a car but, it is becoming clear that regardless of carmaker, new vehicles will be made using LIN for the lowest data-rate functions, CAN for medium speed, MOST (Media Oriented Systems Transport) for the high-speed data rates and FlexRay for safety-critical applications such as steer and brake-by-wire.

II. LIN

A. Local Interconnect Network

The need for a cheap serial network became necessary as the technologies and the facilities implemented in the car increased, while the CAN bus was expensive to be implemented for every component in the car. LIN is a serial network protocol used for communication between components in vehicles. Data within the LIN is transmitted on a single wire serially as eight bit data bytes with one start & stop-bit and no parity. Bit rates vary within the range of 1 kbits/s to 20kbit/s. Usually in automotive application, the LIN bus is connected between smart sensor or actuators and an ECU which is often a gateway with the CAN bus.

B. Working

LIN network comprises of one master and typically up to 16 slaves. All messages are initiated by the master with at most one slave replying to a given message identifier. The master node can also act as a slave by replying to its own messages. Since all communications are initiated by the master, it is not necessary to implement collision detection. A message contains the following fields:

- Synchronization break: Break field indicates a new incoming frame.
- Synchronization byte: The sync field allows the slave to synchronize its clock with the master clock.
- Identifier byte: The message ID field specifies which slave the message is intended for.
- Data bytes: The data field contains the communicated data byte Checksum byte:
- The checksum ensures correct reception of the data frame.

![Figure 1. Master-Slave Communication](image-url)

Two types of messages can be distinguished, as shown in Fig.1[7]:

1020
1) Transmit frame: Contains data sent from the master to the slave. 
2) Response frame: Contains data sent from the slave to the master.

In both cases, the header part containing break, sync, and the message ID is generated by the master. The data part, including the checksum can be generated by either the master or the slave depending on whether it is a transmit or a response frame.

Typically, messages are transmitted at a fixed interval, that fixed interval is called the “tick time”. The LIN master has a schedule table which defines the sequence of messages that will be transmitted on the LIN bus. The LIN master can manage upto 16 slaves on one bus. When the schedule table becomes too long or, if the number of slaves is too large, one LIN master can also control multiple LIN buses.

LIN features a mechanism that allows devices to enter the sleep state and potentially conserve power. All the slave nodes in an active LIN cluster can be changed into sleep mode by sending a diagnostic master request frame with the first data byte equal to zero. This special use of a diagnostic frame is called a go-to-sleep-command. Slave nodes can automatically enter a sleep mode if the LIN bus is inactive for more than 4 seconds. Any node in a sleeping LIN cluster can send a request for wake up cluster. The wake-up request can change the bus to the dominant state for 250 ms to 5 ms. Every slave node can detect the wake-up request (a dominant pulse longer than 150 ms) and be ready to listen to bus commands within 100 ms, measured from the ending edge of the dominant pulse. The master node can also wake up and, when the slave nodes are ready, start sending frame headers to find out the cause of the wake up. If the master does not gain frame headers within 150ms from the wake up request, the node sending the request can try to send a new wake up request. After three failing requests the node shall wait minimum 1.5 seconds before sending a fourth wake up request.

C. Advantages of LIN

1) Low cost silicon implementation based on common UART/SCI interface hardware.
2) In LIN, the synchronization mechanism allows clock recovery by slave nodes without quartz or ceramics resonator. Only the master node will be using the oscillating device. Hence significant cost is reduced.
3) No collision detection exists in LIN, since all messages are initiated by the master with at most one slave replying for a given message identifier.
4) A major benefit of the LIN network is that software development can be avoided. Simple LIN slaves that act as IO extensions in switch modules can be configured quickly and simply by configuring the data E2PROM during end of line programming.

D. LIN drawbacks

1) Lower bandwidth: This is because LIN is a single wire implementation and there is no multiplexing scheme used for the transmission of data.
2) Less effective bus access scheme with the master-slave configuration, as the master alone can initiate communication.

E. LIN v/s CAN

Typically, LIN and CAN cannot be compared as they are used for separate applications. However, to provide a general understanding and to appreciate the simplicity of LIN, a brief comparison of LIN and CAN is as shown in Table 1[1]:

<table>
<thead>
<tr>
<th>Features</th>
<th>LIN</th>
<th>CAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Control</td>
<td>Single Master</td>
<td>Multiple Master</td>
</tr>
<tr>
<td>Maximum Bus Speed</td>
<td>20 Kbps</td>
<td>10 Mbps</td>
</tr>
<tr>
<td>Typical # Nodes</td>
<td>2 to 16</td>
<td>4 to 20</td>
</tr>
<tr>
<td>Message Routing</td>
<td>6-bit Identifier</td>
<td>11/29 – bit Identifier</td>
</tr>
<tr>
<td>Data Byte/ Frame</td>
<td>2,4,8 Bytes</td>
<td>0 to 8 Bytes</td>
</tr>
<tr>
<td>Error Detection</td>
<td>8 – bit Checksum</td>
<td>16 bit CRC</td>
</tr>
<tr>
<td>Physical layer</td>
<td>Single - wire</td>
<td>Twisted pair</td>
</tr>
</tbody>
</table>

Table 1: LIN versus CAN

F. F. Typical LIN Applications

Some typical LIN applications include: Mirrors, window lift, doors switches, door lock, HVAC motors, control panel, engine sensors, engine cooling fan, seat positioning motors, seat switches, wiper control, light switches, interface switches to radio/navigation/phone, rain sensor, light control, sun roof, RF receivers, body computer/smart junction box, interior lighting, and more (Fig.2[6]).

III. FLEXRAY

FlexRay is a scalable, flexible, high-speed, deterministic, error-tolerant communication technology that is designed to meet growing safety related challenges in the automobile industry. This technology is mainly concentrated for data communication in very safety critical use areas in automobile.
The communication cycle as shown in Fig.4, which is periodically repeated, is composed of static and dynamic segments, symbol window and NIT (network idle time). Within the static segment the TDMA (time division multiple access) method is used. This segment is usually designed for the high priority frames with the same length of payload. It is divided into static slots where each node can transmit its own frame. Several static slots can be assigned to one node, but each of these static slots may be used just by one node. The FlexRay dynamic segment is intended for the non-critical and sporadic messages with variable length. An access to the bus is based on the flexible TDMA. In the symbol window the CAS (Collision Avoidance Symbol), MTS (Media Access Test Symbol) or WUS (Wake Up Symbol) can be sent.

For safety critical applications, messages can be transmitted simultaneously on both channels giving a built-in redundancy to the network. If one channel gets damaged, transmission will continue without interruption on the other channel. This is essential if drive-by-wire applications are to be implemented, so that, in the event of a failure on a channel the driver would still be able to have full control over the vehicle’s brakes and steering. Data is transmitted on the FlexRay bus in both timed and event driven manner. Each message is divided into static segment and the dynamic segment. The static segment is defined during the configuration of the application and transmits the data on a TDMA basis. The dynamic segment of the message handles data on an event triggered basis. It can be configured as a single-channel or dual-channel bus network, each node on the network can be connected to either or both of the channels. This flexibility in configuration may be used to increase bandwidth and/or introduce redundancy in to the system to increase its level of fault tolerance. The dynamic segment is used for event based messages; here the devices compete for and width using a priority driven scheme. The size of the communication slots in the dynamic segment may vary to accommodate frames of different length, but data will only be sent if there is enough time left in the dynamic segment.

A FlexRay communication system (Fig.3[2][3]) is made up of a number of FlexRay nodes and a physical transmission medium (FlexRay Bus) interconnecting all of the FlexRay nodes. The FlexRay node is an ECU which is connected to a FlexRay bus via a FlexRay interface. FlexRay interface is made up of communication controller and one or two bus drivers depending on the number of channels. Basically there are two channels available. System designer can choose between single channel or dual channel configuration. The main physical topologies are bus, star and ring. The hybrid topology of the mentioned basic topologies is also possible.

FlexRay manages multiple nodes with a Time Division Multiple Access or TDMA scheme. Every FlexRay node is synchronized to the same clock, and each node waits for its turn to write on the bus. Because the timing is consistent in a TDMA scheme, FlexRay is able to guarantee determinism, or the consistency of data deliver to nodes on the network. This provides many advantages for systems that depend on up-to-date data between nodes.

The communication cycle as shown in Fig.4, which is periodically repeated, is composed of static and dynamic segments, symbol window and NIT (network idle time).
C. CAN v/s Flexray

A brief comparison between CAN and FlexRay is presented below.[5]

1) Complexity: The CAN system is considered to be a low complexity system due to the single channel bus with arbitration method for bus access. The FlexRay is considered to be very complex due to use of two different communication accesses. FlexRay uses both static and dynamic bus access which makes the whole system very complex.

2) Safety: The CAN protocol has many features to ensure safety such as bit monitoring, CRC and fault confinement. Its broadcast bus with prioritization of messages ensures deadlines for high priority functions. The CAN bus uses line topology greatly reducing the risk of the system. Since FlexRay is supposed to be the next generation communication system for automotive networks, safety is one of the major motives in its development. The systems may be configured in many ways with the two channels, so if one fails the system could keep its functionality. Each node in the FlexRay system is assigned a bus guardian which is a safety measure to ensure that the node does not get stuck in an erroneous mode.

3) FlexRay has the ability to make the system both static and dynamic with regards to the handling of messages. This makes it easy to add new nodes without having to change the scheduling of the system. Nodes can therefore be tested separately and then integrated into an existing system. Whereas nodes of CAN network cannot be tested separately.

4) Flexibility: The CAN system provides some form of flexibility but restricts in many ways. Like the name, FlexRay is intended to be very flexible system. Due to its two channels it may be configured in different ways like bus, star and hybrid topologies. The standard design of ECU interface makes the system highly flexible.

D. FLEXRAY applications

Some typical examples of FlexRay X-by-Wire applications[2] include:

- Steering-by-Wire—Typically using electronic control unit
- Anti-lock brake system (ABS)—Including vehicle stability control (VSC) and vehicle stability assist (VSA)
- Power train—Controlling an electronic throttle that replaces the current mechanical system. The electronic throttle works in conjunction with existing systems such as a computerized fuel injector, computerized variable intake control system, and computerized idling control system.

IV. CONCLUSION

LIN (Local Interconnect Network) is the communication protocol for a low-speed network to which all comfort and convenience applications in the car are connected. LIN can be used for every application which has no demand for high data rates. LIN has the potential to cover all applications which cannot be directly connected to CAN because of high cost. FlexRay provides immense flexibility to the designer and is used for critical priority applications. However, FlexRay is a complex concept. As of today, FlexRay is applied only for safety related applications.

REFERENCES