

LIN (Local Interconnect Network):

History:

LIN (Local Interconnect Network) was developed as cost-effective alternate to CAN protocol. In 1998 a group of companies including Volvo, Motorola, Audi, BMW, Daimler Chrysler, and Volkswagen formed a consortium to develop LIN.

The latest version of LIN is LIN 2.0 and was released in 2003. LIN 2.0 provides interesting features such as diagnostics.

Introduction:

The LIN is a SCI/UART-based serial, byte-oriented, time triggered communication protocol designed to support automotive networks in conjunction with Controller Area Network (CAN), which enables cost-effective communication with sensors and actuators when all the features of CAN are not required. The main features of this protocol (compared to CAN) are low cost and low speed and used for short distance networks.

Usually in automotive application, the LIN bus is connected between smart sensor or actuators and an Electronic Control Unit (ECU) which is often a gateway with CAN bus. Like CAN, LIN is also a broadcast type serial network, but with single master and multiple (up to 16) slaves. No collision detection exists in LIN, therefore all messages are initiated by the master with at most one slave replying for a given message identifier. The master is typically a moderately powerful microcontroller, whereas the slaves can be less powerful, cheaper micro-controllers or dedicated ASICs.

Moreover the LIN is a single wire 12V bus connection, in which the communication protocol is based upon ISO 9141 NRZ-standard. An important feature of LIN is the synchronization mechanism that allows the clock recovery by slave nodes without quartz or ceramics resonator. Only the master node will be using the oscillating device. Nodes can be added to the LIN network without requiring hardware or software changes in other slave nodes. And the maximum transmission speed will be 20kbit/s.

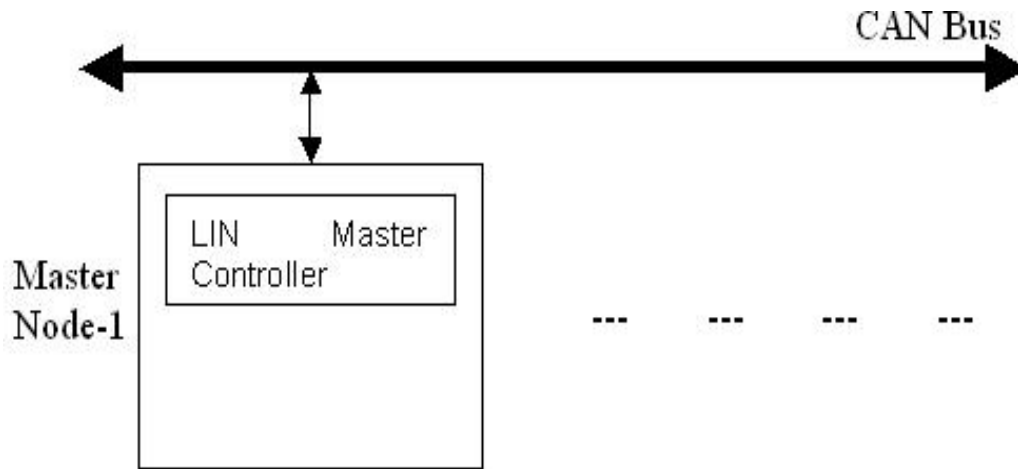


Fig 1-a

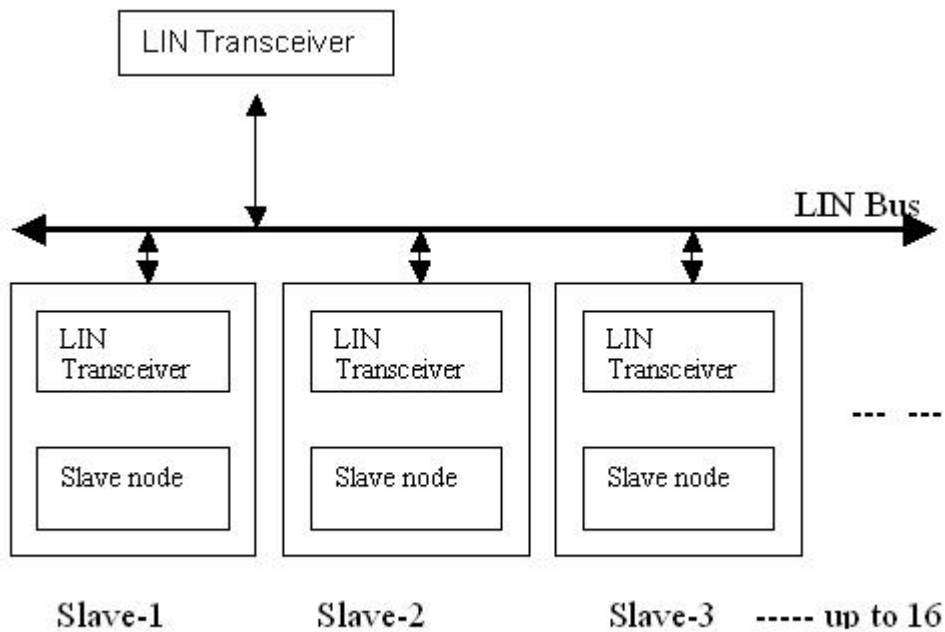


Fig-1 LIN network overview

How do they transmit & receive data?

As we have already seen, the LIN network comprises of one master node and one or more slave nodes. Only the master node initiates the communication in a LIN network. The master node defines the transmission speed, sends synchronization pulses, and does data monitoring and switching slave nodes to sleep/wake up mode. It also receives Wakeup Break from slave nodes when the bus is inactive and they request some action.

A slave node waits for the synchronization pulse, and then does synchronization using synchronization pulse and process the message identifier. Then according to the ID, the slave decides what to do, either to receive data or to transmit or to do nothing. While transmitting it sends 2, 4 or 8 data bytes depending on the ID received, plus a checksum byte. The two types of data messages in the LIN network are the signal message (data which are sent in the data frame) and the diagnostic message.

The network consists of one master task and several slave tasks. In the master node the both tasks-master & slave can be found, but in a slave node the slave task only.

The communication can take place from the master node (using its slave task) to one or more slave

nodes, and from one slave node to the master node and/or other slave nodes. The communication is also possible directly from slave to slave without routing through the master node. The LIN uses frames for data communication. A frame consists of a header, a response and some response space so the slave will have time to answer. The master sends out a message header, or in other words the headers are situated in a master task, contains synchronization breaks, synchronization byte and the message identifier, each part begins with a start bit and ends with a stop bit. The response contains one to eight data bytes and one checksum byte. The slave task is connected to the identifier and receives the response, verifies the checksum and uses the data transport. Messages are created when the master node sends a frame containing a header. The slave node(s) then fills the frame with data depending on the header sent from the master.

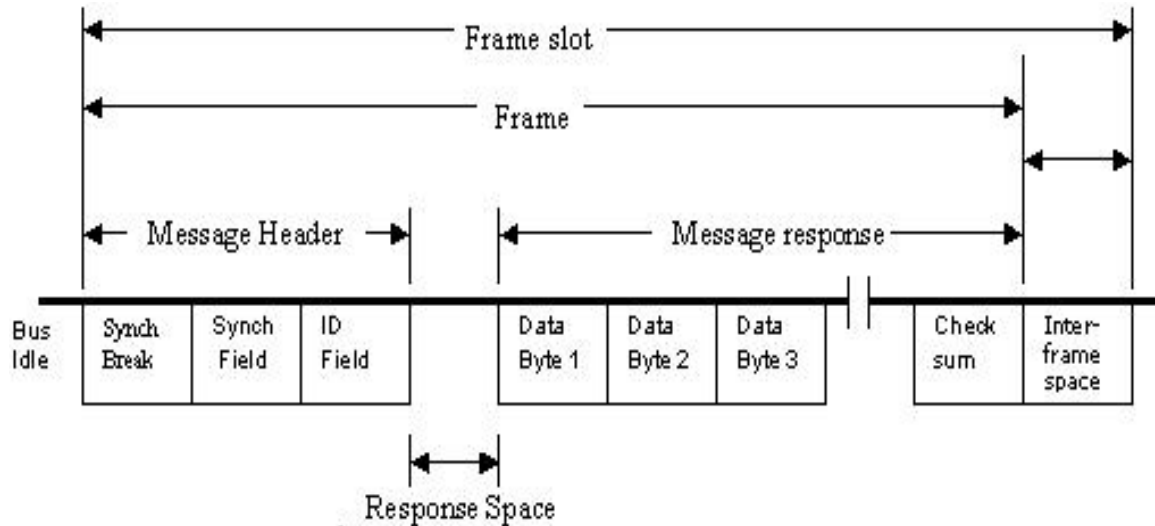


Fig-2 LIN frame

- This system of communication using headers and responses has a lot of advantages. Some of them are
1. The nodes can be added to the network without requiring hardware or software changes in other slave nodes.
 2. The identifier defines the content of a message and
 3. Any number of nodes can simultaneously receive and act upon a single frame.

The LIN protocol is byte oriented, which means that data is sent one byte at a time. One byte field contains a start bit (dominant), 8 data bits and a stop bit (recessive). The data bits are sent LSB first. The synch break is the beginning of a Message Frame. It contains at least 13 bits of dominant value including the start bit followed by at least 1 bit long recessive break delimiter. The synch byte or the synch field is used to determine the time between two rising edges to determine the transmission rate which the master node uses.

The identifier of a message, which is 6-bits long identifier plus 2 parity bits, denotes the content of a message but not the destination. It incorporates information about the sender, the receivers, the purpose, and the data field length of the response (The three classes of 2/4/8 data bytes). The length coding is done in the 2 MSB of the ID-Field. A total of 64 message Identifiers is possible here. The two linked parity bits are used to protect this ID-field.

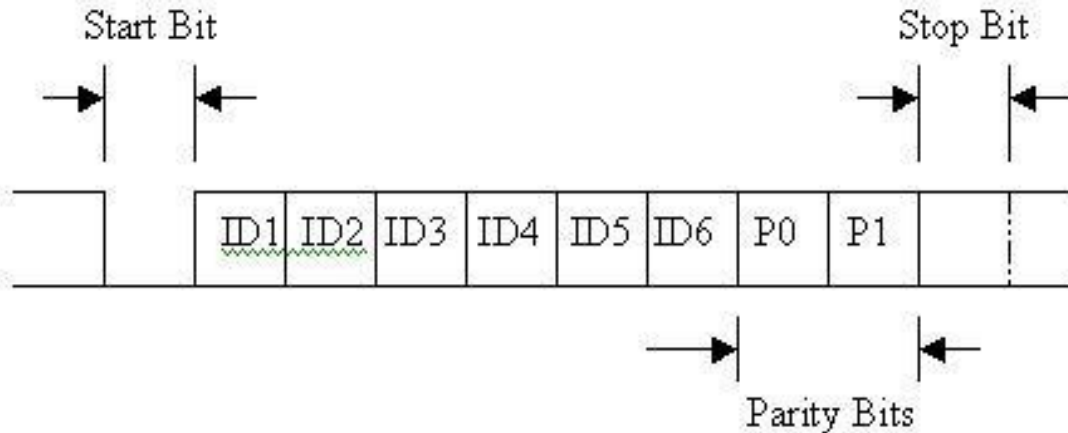


Fig-3 Identifier field

(Fig 2)

The parity bits can be calculated as explained below:

P0 = logic OR between ID0, ID1, ID2 and ID4.

P1 = logic OR between ID1, ID3, ID4 and ID5.

The length of the response data field from the slave can be 2, 4 or 8 bytes, depending on the two MSB of the ID field send by the master node (LIN 1.3 have length of 8-bytes only)

There are two types of checksum used in LIN. The first one used in LIN 1.3 consists of the inverted eight-bit sum of all 8 data bytes in a message. The new checksum used in LIN 2.0 also incorporates the protected identifier in the checksum calculation.

The power management in LIN network:

The network management in a LIN cluster contains "wake up" and "go-to sleep".

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 150 ms 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.

LIN Versus CAN:

There is no meaning of comparing CAN and LIN since they do not address the same issues. However it can give you a general view of LIN in the big picture.

As we already saw, the LIN targets low-end applications where the communication cost per node must

be two to three times lower compared to CAN but where the performance, robustness and versatility of CAN are not required. The main economical factor in favor of LIN is the avoidance of high cost quartz or ceramic resonators in slave nodes, as they can perform self-synchronization. A small comparative study gives the following points.

Features	LIN	CAN
Media access control	Single Master	Multiple Masters
Bus Speed	2.4, 9.6 and 19.2 Kbps	62.5Kbps to 1Mbps
Multicast Message Routing	6 bit identifier	11/29 bit identifier
Size of network	2 to 16 nodes	4 to 20 nodes
Data Byte per Frame	0 to 8	2 to 8
Transmission time for 4 data bytes	6 ms at 19.2Kbps	0.8 ms at 125Kbps
Error detection(Data field)	8-bit checksum	15 bit CRC
Physical Layer	Single wire, 40V	Shielded, Twisted pair, 5V
Quartz/ ceramic Resonator	For master only	For All nodes

Some of the main drawbacks of LIN are lower bandwidth and less effective bus access scheme with the master- slave configuration.