1.5 Local Interconnect Networking

Local interconnect network (LIN) is a communication protocol that has been established for automotive vehicles [KOPETZ ET AL. 2003]. This protocol is based on the SCI (UART) data format with single master/multiple slave architecture. A consortium formed in 1998, comprising AUDI, BMW, DaimlerChrysler, Volvo, Volkswagen, VCT, and Motorola worked on establishing a specification for this protocol. The development of LIN was based on the necessity for a communication protocol that was very cost effective and not only addressed the issue of the specification for communication, but also other issues like signal transmission, programming, and interconnection of nodes. It basically takes an all-round approach for the development and consolidation of an automotive protocol.

The basic advantage that LIN enjoys is that it is very economical compared to protocol such as CAN (Fig. 1.27). However, this advantage is negated by its inherent limitations, such as low bandwidth and performance, and the single master topology of the network [SHRINATH AND EMADI 2004; MOTOROLA 2007].

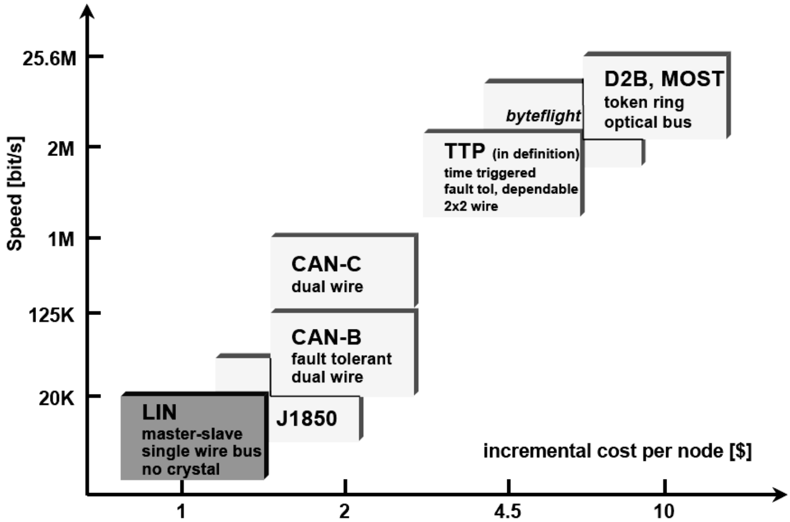

Fig. 1.27 MUX Standards (Costs and Data Rates) [MOTOROLA 2007].

There are quite a few criteria that have to be taken into consideration when a network is designed. Factors such as bandwidth, security, latency, **electromagnetic interference** (EMI), fault tolerance, and cost should be balanced in order to design a network to suit requirements. There are basically two different approaches that could be taken to design a network.

B.T. Fijalkowski, *Automotive Mechatronics: Operational and Practical Issues*,
Intelligent Systems, Control and Automation: Science and Engineering 47,
DOI 10.1007/978-94-007-0409-1_5,

One approach is to divide the sensors and actuators in a zonal manner, connected to a central ECU. Various main ECUs are connected to each other by CAN links. This extensive use of CAN is to enable the usage of high-bandwidth for signal exchanges. The other approach is to totally abolish the zonal concept. Here, all the actuators and sensors are connected to the central ECU by means of LIN links. This has the advantage of being scalable. No major changes have to be done to the network in order to accommodate additional nodes. As mentioned before, LIN is based on the SCI (UART) byte word interface. The network has a single master/multiple slave topology [SPECKS AND RÁJNAK 2000].

All the slaves have the job of transmitting and receiving. The master node, apart from the task of transmitting and reception, has the additional task of maintaining the synchronisation in the network. This is done by means of a message header that consists of the synchronisation break, synchronisation byte, and a message identifier. The message identifier is used by the nodes to identify the messages meant for them. Each identifier is unique to the node. This way the node knows the messages meant for it. It has to be remembered that the message identifier indicates the content of the message and not the destination. Once the node has reached the message meant for it, it then sends back a response that contains data with the size ranging from 2, 4 or 8 bytes of data with one checksum byte. One message frame consists of the header and the response parts [SHRINATH AND EMADI 2004].

Each message frame is made up of a byte field that has 10 bits. These 10 bits include a dominant *'start'* bit, 8 bits of data and recessive *'stop'* bit. The message frame consists of the header sent by the master node and the response sent by the slave node. The header sent by the master node consists of a synchronisation break, synchronisation field, and the identifier field. Each message frame is built on the *8N1*-coding scheme. The synchronisation break must be a minimum 13 bits in length to ensure proper synchronisation. The synchronisation field is a string of bits with an equivalent hexadecimal value of 0×55. With this type of synchronisation pattern, it is possible for nodes not equipped with quartz stabilisers to resynchronise themselves. There are basically two levels of synchronisation that are defined for the LIN: unsynchronised, in which the slave clock time differs from the master clock time by less than $\pm$ 15 % and synchronised, in which the slave clock time differs from the master clock time by less than $\pm$ 2% [SHRINATH AND EMADI 2004]. There are four identifiers reserved for special purposes. Out of the four identifiers, two are used for uploading and downloading purposes. The only way they differ from the normal data frame is that in these frames, instead of data, there are user-defined command messages. The other two special identifiers are used for ensuring upward compatibility with the future versions of the LIN protocol. These identifiers are termed *'extended identifiers'*. There is three types of communication mode there are supported: master to slave or multiple slaves, slave to master, and slave-to-slave: the slaves may talk each other without routing the transmission through the master. This is illustrated in Figure 1.28 [SHRINATH AND EMADI 2004]. A typical LIN has a single master with as many as 2 -- 10 nodes contained in the network. Typical data speeds range from 2.4 to 19.6 kb/s. The voltage level that may be supported by the buses is around 13.5 V_{DC}.

Each data frame contains 2 -- 8 bytes of data with a six-bit identifier. Single-wire transmission, low cost of implementation, and no resonators are the advantages. Low bandwidth and a single-bus access scheme are the main disadvantages.

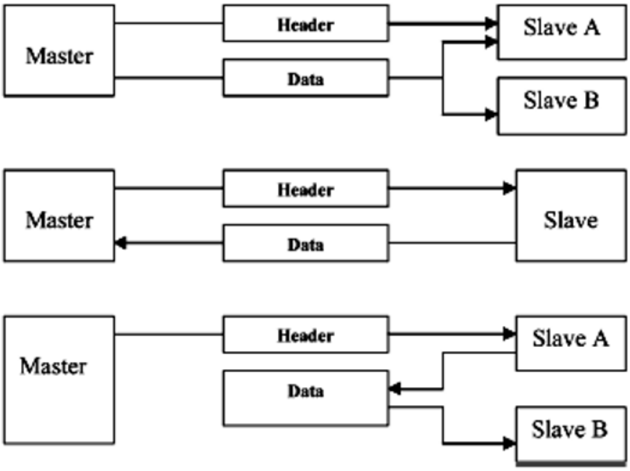

Fig. 1.28 **Data** communication in LIN [SHRINATH AND EMADI 2004].

LIN is usually used for door and roof control, as well as for the steering HW and column (Fig. 1.29) [MOTOROLA 2007].

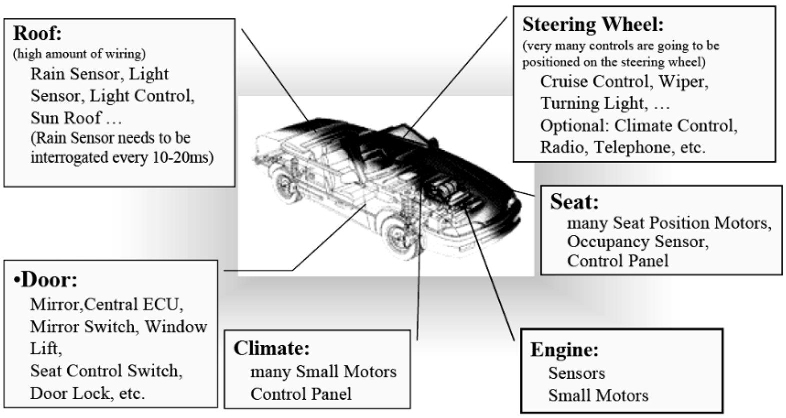

Fig. 1.29 Classic LIN applications [MOTOROLA 2007].

Other implementations include smart wiper E-M motor, sensors and switch panels, seat control, and heating [SHRINATH AND EMADI 2004; MOTOROLA 2007].