



# LIN BUS Formulation using Optical Fiber

Vivek P Kude<sup>1</sup>, Shubhangi B Patil<sup>2</sup>

Department of Applied Science, MGMs College of Engineering, Nanded, India

<sup>1</sup>Department of Electronics, <sup>2</sup>Dr. J J Magdum College of Engineering, Jaysinghpur, India

Corresponding Author: Email: [kude\\_vp@mgmcen.ac.in](mailto:kude_vp@mgmcen.ac.in)

## How to Cite this Article:

V Kude, V. P. & Patil, S. B. (2026). LIN BUS Formulation using Optical Fiber. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(03). <https://doi.org/10.55041/ijcope.v2i3.105>

## License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i3.105>

## Abstract:

The maximum automobiles may use the various types of sensors in new advanced vehicles for as per the consumer's requirements. These sensors may connect using local interface network (LIN). The sensor interfacing of the network is carrying out using copper wired connections for every automobile. The copper wired connections may take time latency for signal transmission, also it generates electromagnetic immunity and cross talk, in emergency case this may give obstruct to user. In the direction of to reduce and avoid such instances, in our work we proposed local interface network using optical fiber. As previously researchers proved that optical fiber have very minute electromagnetic immunity and cross talk also it has high speed signal data transmission capacity as compared with copper wired connections. On the given basis we have observed results which shows that LIN message transfer rate using optical fiber is 19.2 Mbps whereas using wire it is 19.2 Kbps. The LIN message stack period using wire is 19.2 milli seconds and using optical fiber it is 19.2 micro seconds.

Key words: LIN, Optical fiber, fuel sensor, radiator sensor, electromagnetic



## I. Introduction:

The LIN network is implemented using single wire, which generally means higher values of EME (Electromagnetic emission) compared to differential twisted pair-wire implemented such as CAN. To keep the EME values low, the slew rate of the signals and also the bandwidth of the data transferred is also controlled [1]. As per LIN specification the data rate may not exceed above 20 Kbits/s. The configuration of LIN network consists of a single master and one or several slaves (up to 16 is recommended by the LIN specification [LIN02]). The master initiates all data transfer on the bus and did communication with each slave available on network. Similarly, each slave may communicate the message data frame to master using the same bus [2]. This leads to deterministic traffic behavior and guarantees the latency times calculated for the specified frames transmitted on the LIN network. This will limit the transmission length up to 40 meters.

## II.I LIN message frame

The LIN message frame consists of a header and a response part. The header has a fixed length while the response part consists of 0 to 8 bytes of data. The inter-frame-response time is the time it takes for a slave to respond to a request (i.e. to a ID) from the master and it may vary among the nodes on the network since it depends on the hardware and software implementation in each node [3]. At the end of the response part a checksum, which is calculated for the data part, is attached. The header is broken up into three fields: the SYNC-break, SYNC-field and the identifier- (ID) field. The structure of the message frame can be seen in Figure 1

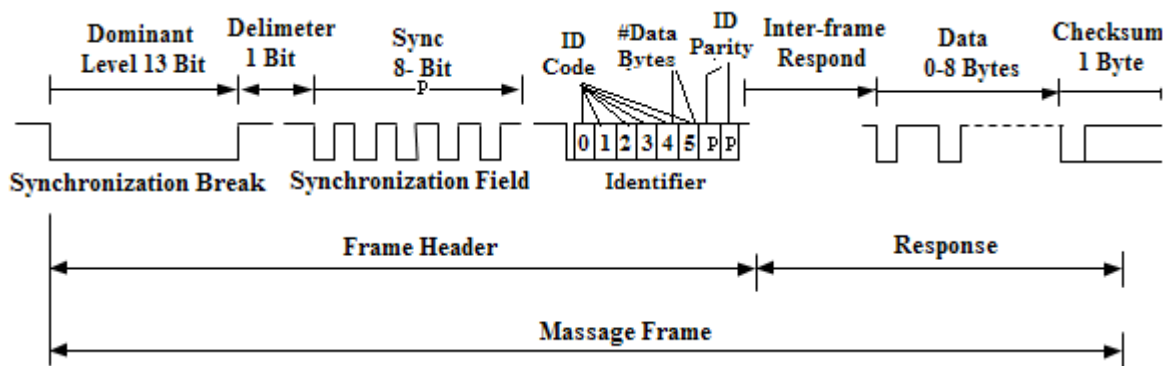


Figure 1: The structure of the message frame

## II.I.I Synchronization break

The first part of a LIN-message is the SYNC-break, which consists of at least 13 bits of zeroes. The break is needed in order for the slaves to detect that a message is transmitted on the bus. According to the specification [LIN02] the slaves are allowed to have a clock frequency (i.e. baud rate) that differs with 15% to the masters. With this in mind, for a unsynchronized slave node with a clock frequency that is 15% slower than the master's clock to detect a break, it has to sample at least 10 bits as zeroes since 9 zeroes can still be found in an ordinary UART-byte. For the master this would mean that  $10 \times 1.15 = 11.5$  bits would be sufficient to send in order for the slaves to detect the break. However, the specification states that a minimum of 13 bits should be sent [4].

For a microcontroller with a UART-interface this means that a framing error is flagged when a SYNC-break is received since the stop bit in the UART-byte is sampled as a zero instead of a one. Further, the LIN-software routines in the slave have to check that all the bits in the data byte received are zeroes in order to be sure that a break has been received [5].

For the master node, implemented in a microcontroller, the procedure of sending a SYNC-break also involves some tampering with the UART-protocol (assuming that the UART-module is used) since it is impossible to send 13 bits of zeroes in a row. One way is to change the baud rate to a slower rate so that the time taken to send a UART-byte corresponds to 13 bits with the right baud rate.

### II.I.II. Synchronization field

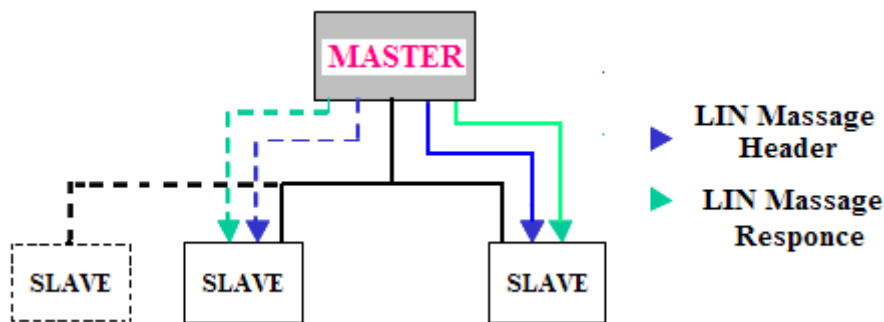
Since the master always initiates a transmission by sending a header, the slaves are able to synchronize their clocks every time a new message is received. This makes it unnecessary to implement expensive resonators or oscillators on the slave nodes and only one, by comparison, accurate resonator is necessary in the master as a time reference [6].

The slave synchronizes its clock by measuring the time taken from the first falling edge of the ID-field (i.e. the falling edge of the start bit) to the fifth falling edge i.e. bit 7 of the SYNC-byte and divides it by 8 in order to get the bit time or baud rate of the master.

### II.I.III Identifier field

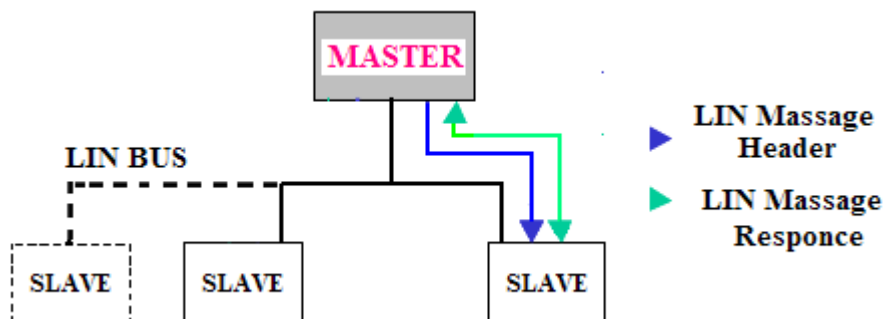
In the last part of the message header the ID-field is placed. The ID-field denotes the content of the data part of the message and is protected with two parity bits. In the LIN- specification rev.1.2 (see [LIN00]), two of the bits in the ID-field are used for specifying the length of the data part of the message and the number of bytes allowed were 2, 4 or 8. In the current revision this is no longer the case and 0 to 8 bytes of data can be used.

The slave nodes on the network are addressed by the ID-field. The nodes do not have physical addresses (e.g. MAC-address or any other hardware address) but instead use a pre-programmed list of valid IDs in memory that they use to filter out which messages to respond to. This way the ID can have different meaning for different nodes and it enables 3 ways of passing data between the master and its slaves [7].



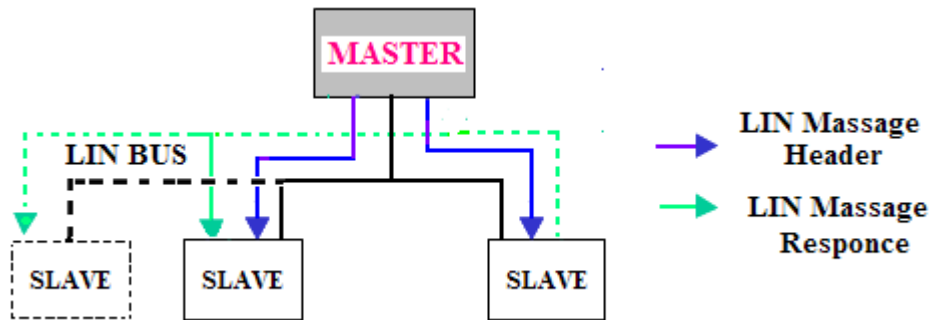
**Figure 2: Data from Master to Slave (s)**

Master – slave(s) communication, see Figure 2. The first alternative of passing data is when the master sends out a complete message frame i.e. both header and response for one or more slaves (multicast if more than one or broadcast if all slaves listen for it) and is usually referred to as a command frame (CMD frame).



**Figure 3: Data from Slave to Master**

Slave – master communication, see Figure 3. This alternative is conducted when the master requests a response from one specific slave (polling) and is usually referred to as request frames (REQ frame).



**Figure 4 Slave – slave(s) communication**

Slave – slave(s) communication, see Figure 4. The last alternative is when one slave sends its response to one or more slaves and can be used when there is no need for data processing by the master e.g. information sent between sensor and actuator.

#### III.I. IV. Error Detection, fault confinement and data protection

The actions taken by the nodes on the network when a message gets corrupted is not specified by the LIN-specification and therefore it is up to the application layer to take care of the fault confinement procedures. The LIN-protocol only specifies basic error detection schemes for bit-, checksum-, ID-parity-, slave not responding-, inconsistent synch field- and no bus activity-errors [8]. The fault confinement is taken care of by the master, since it is the only node that can issue a re-scheduling of a message. Slave nodes cannot directly signal errors; therefore, the master has to poll the slaves for errors. Bit-errors at the transmitter (at both master and slaves) are detected by comparing the outgoing message stream with the monitored message stream. Checksum- and ID-parity-errors are easily detected by locally calculating and comparing data. The inconsistent-synch-field-error is detected when the edges of the synch-field are outside the given tolerance. The slave not responding- and no bus activity are easily detected by use of timers. If a slave node observes an inconsistency, it can save this as diagnostic information, which it can relay to the master when requested.

#### III.I.V. Scheduling

The master on a LIN-network handles all transmissions of frames according to schedules. The schedules are built-up with respect to the real-time requirements of the signals included in the frames. Each frame takes up a certain amount of time in the schedule and is called frame-delay. It is the time given for a frame to be transmitted before next frame in the schedule is transmitted. The actual time taken for a frame to be transmitted is less than the frame-delay and therefore some slack time is introduced in-between the entries of the schedule [9]. The LIN-specification supports the use of multiple schedules, which can be useful in many cases [10]. One example is the door-module with various functions such as motorized window-lift, motorized controlled rear-view mirror with heating, lock etc. of a vehicle. In this example an extra schedule could be useful when the up-button of the window-lift function is pressed. The master switches to a schedule that prioritizes the messages sent to and from the window-lifts motor in order to cut down the response-time for when the button is released. This is important since you do not want any delays for this event, e.g. a finger could be pinched.

#### III.I. LIN Real-time properties

Since the communication between nodes on the network is governed and controlled by the master, there are relatively little real-time conflicts that can occur on the bus. There is no bus arbitration and therefore no collisions will ever occur (except when something goes very wrong), which excludes retransmissions and makes the transmissions very predictable. Despite the deterministic behavior there are some factors that are important when designing and scheduling frames and signals for the network. Some of these factors concern signal latencies, see Figure 5, that are inflicted in various points in the communication and which effects the real-time properties of the system [11].

### II.II.I. Signal latencies

The time-model in Figure 5 shows six time points between generation and consumption of a signal value. Note that this model describes a single consumer. There can be several consumers of a single signal on the network and the values of the described latencies are specific for each node. This is because the system might have different kinds of software and hardware solutions for the nodes in the system.

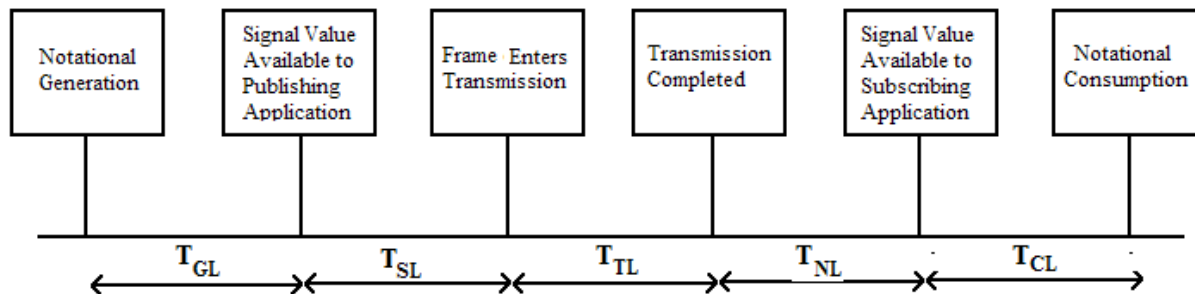


Figure 5–Signal and frame latencies

#### Generation latency, $T_{GL}$

Defines the time taken from that an event has occurred (e.g. a button is pressed) to that the corresponding signal is updated in buffer (by the publishing application) and available for the LIN-drivers for transmission. This is a latency that depends on the properties of the publishing application (e.g. what kind of hardware and software that is used) and hence does not affect the real-time properties of the LIN-protocol.

#### Schedule latency, $T_{SL}$

Defines the time taken before the actual transmission is done and depends on when the frame with the signal is due for transmission and is determined by the master's schedule. The latency is the same for all signals in the frame and also common for all subscribers of the signal. This latency depends on how the scheduling of the frames is done, if it's a tight schedule or if it uses a lot of slack time between frames for other task executions on the microcontroller.

#### Transmission latency, $T_{TL}$

$T_{TL}$  is the time required for transmitting a frame on the bus and depends on the speed of the transmissions and the length of the bus.

#### Notification latency, $T_{NL}$

Defines the time taken between the reception of the updated signal and the subscribing application has stored the value in buffer. Is like  $T_{GL}$ , dependent of hardware and software implementations and does not affect the real-time properties of the LIN- protocol.



## Consumption latency, $TCL$

Defines the time from that the application is notified by the updated signal, to that the corresponding action is performed by the application. Like  $TNL$ , dependent on the hardware and software and not specific for the LIN-protocol.

### III.II. Frame delay and time base

Each frame transmitted on the bus is given a certain amount of time for completing its transmission before the next frame in the schedule table is started. This is called the frame delay and is not necessary the same for all frames in the schedule. The delay depends on the size of the frames and also the time base of the system.

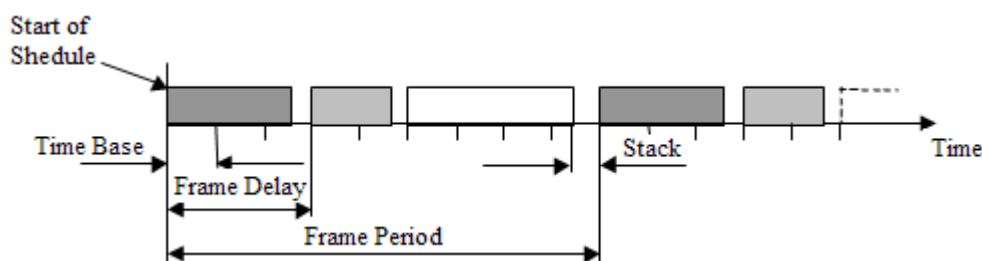


Figure 6: The concept of frame delay and time base

In some systems it's perhaps mandatory to have great time-gaps between frames so that other tasks on the nodes (both master and slave) are given more time to execute. On other systems the schedule needs to be as tight as possible due to tight real-time demands.

The time base of the system gives the resolution of the frame delay in milliseconds. With a smaller time base the frame delay can be more tightly formed to the actual time constituted by the frame i.e. minimizing the slack between frames. The time base is not a value specified by the LIN-specification but is something that the system developer configures. Figure 6 shows the concept of frame delay and time base. An example in complement to the figure can be given:

A frame  $F$  consists of 2 bytes of data. The maximum bit length of the frame gives the maximum time within which the transmission of the frame must be completed. It is calculated according to the LIN-specification as:

$$\text{Length}_{\max}(F) = \text{Length}_{\min}(F) * 1.4$$

$$\text{Length}_{\max}(F) = [(10 * n_{\text{data}}) + \text{Length}_{\min}(\text{header}) + \text{Checksum Byte} + 1] * 1.4$$

$$\text{Length}_{\max}(F) = [(10 * 2) + 34 + 10 + 1] = 91 \text{ Bits}$$

As one can see, the minimum length of a frame is extended 40% to give the maximum allowed length. This is done so that the subscribing frames will have some inter-frame response time for calculations and other tasks.

If the speed of the transmissions using wire is set to 9.6kbits/s and 20 Kbits/s the maximum response time is calculated to:



$$\text{Max Time} = \frac{91}{9600} = 9.5 \text{ ms}$$

$$\text{Max Time} = 91/20000 = 4.5 \text{ ms}$$

With plastic optical fiber we can limit the speed of data transfer 2 Mbps/s to 20 Mbps/s

Now, if the time base of the system is configured to 5 ms one can see that the delay of the frame  $F$  would be a minimum of  $2 * 5 \text{ ms} = 10 \text{ ms}$ . The 0.5 ms left before the next entry in the table, is slack time for the master.

The time base is often implemented as a timer that gives a “tic” (e.g. interrupt) every time a time base period has elapsed. The use of a time base is a recommendation given by the LIN specification as a way of implementing the scheduling functionality of the master.

### III.III. Signal Latency improvement using Plastic Optical Fiber.

Transmission Latency is the time required for transmitting a frame on the bus and depends on the speed of the transmissions and the length of the bus [12]. If we may use wire the transmission latency may take the time for to transfer a frame for 9.5 Kbps is 9.5 ms. For an urgent decision this time may very long . To overcome this draw back in our system we are using Poly-Mithyl Metha Acrylate (PMMA) optical fibers [13]. This may improve transmission latency. In order to increase the transmission distance and reduce noise effectively, a communication system with the optical fiber interface is developed in this work

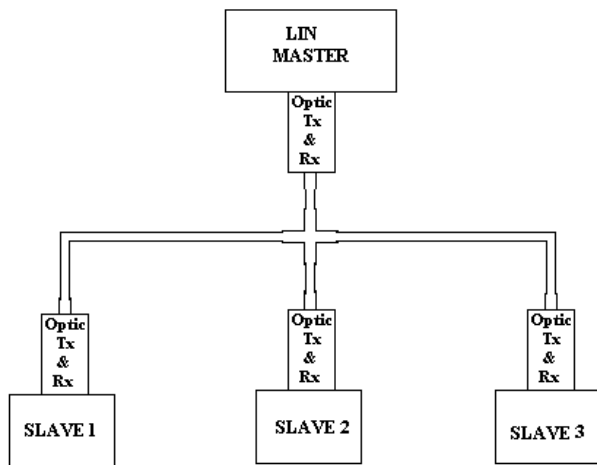


Figure 7: Modified optical fiber LIN communication system with SLAVES

There are several technologies that can serve as the transmission medium for transmission of data such as twisted copper pairs and copper coaxial cables, silica single mode fiber, and multimode glass fiber based technologies [14]. Let us examine the advantages and disadvantages of POF with respect to each of those technologies. Compared to copper based technologies like coax cables and twisted pair, POF guarantees electromagnetic immunity and absence of crosstalk. POF has smaller volume, it is less bulky, more flexible, and it has smaller weight. With respect to data transmission capability, POF offers higher bandwidth at longer transmission distances and it offers lower installation and maintenance costs particularly in splicing[15]. Compared to multimode glass optical fiber, POF is more flexible and ductile making it easy to handle. POF termination can be realized faster and cheaper than in the case of multimode glass fiber. The typical large



core of polymer fiber allows for large tolerance on axial misalignments, which results in cheaper connectors. For comparisons sake, let us examine the power loss due to lateral (axial) misalignment of connecting two graded index (parabolic case) multimode fiber with different core diameters. Calculations, assuming uniform modal power distributions, for a misalignment of 25 microns yield a loss of 1.76 dB for a 62.5 microns core diameter multimode fiber (MMF). For the case of PF-POF with a core diameter of 200 microns the 25 microns displacement results only in 0.48 dB loss. These calculations are based on the theory of power coupling of two PF-POF fibers. Another advantage of large core PF POF has been observed with respect to the bandwidth degradation due to modal noise at misalignments in fiber-to-fiber connections. Namely, PF-POF shows very short time delay at the wide area of the core. This is in contrast to the case of multimode silica fiber with 50 to 62.5 microns core where small displacements cause severe bandwidth degradation.

### III.III.I. Design consideration for Transmission Latency

The packet transmission time ( $P_t$ ) is given by the ratio of packet size ( $P_s$ ) to the Bit Rate ( $B_r$ ).

$$P_t = P_s / B_r$$

Propagation delay ( $P_d$ ) is the time it takes for the first bit to travel from the sender to the receiver (during this time the receiver is unaware that a message is being transmitted). The propagation speed ( $P_v$ ) depends on the physical medium of the link (i.e. fiber optic, copper wire, etc) and is the approximate range of  $2 \times 10^8$  m/sec for copper and  $3 \times 10^8$  m/sec for optical fiber, the propagation time delay ( $P_d$ ) is given by

$$P_d = D_t / P_v$$

Where  $D_t$  is distance travelled by data bit

The packet delivery time ( $P_{dt}$ ) is the time from when the 1<sup>st</sup> bit leaves transmitter until the last bit received in the case of physical link, it can be expressed as

$$P_{dt} = P_t + P_d$$

The round trip time ( $R_{tt}$ ) or ping time is the time from the start of the transmission from the sending node until a response is received at the same node. It is affected by packet delivery time as well as data processing delay ( $D_{pd}$ ), which depends on the load on the responding node. If the sent data packets as well as response packet have the same length, the round trip time can be expressed as

$$R_{tt} = 2 * P_{dt} + D_{pd}$$

In case of only one physical link, the above expression corresponds to link round trip time ( $LR_{tt}$ )

$$LR_{tt} = 2 (P_{dt} + P_d) + D_{pd}$$

If the response packet is very short the link round trip time the link round trip time can be expressed as close to

$$LR_{tt} \approx P_{dt} + (2 P_d) + D_{pd}$$

The network throughput ( $N_t$ ) of a connection with flow control, for example a TCP connection, with certain window size ( $W_s$ ) or buffer size can be expressed as

$$(N_t) = (W_s) / R_{tt}$$

In case if only one physical link between the sending and transmitting nodes this corresponds to Link network throughput ( $LN_t$ )

$$(LN_t) \approx L_b * (P_{dt} / LR_{tt})$$

Where  $L_b$  is the link budget

The message delivery time or latency ( $L_{at}$ ) over a network depends on the message size in bit and network throughput or effective data rate in bit/s as

$$L_{at} = P_s / LN_t$$

The Lin bus is a pulled bus, the processing of each frame is allocated time slot as follows



$T_{header\ Nominal} = 34 * T_{Bit}$   
 $T_{response\ Nominal} = 10 * (N_{data} + 1) * T_{Bit}$   
 $T_{frame\_Nominal} = T_{header\_Nominal} + T_{response\_Nominal}$   
 Processing of each frame is allocated a maximum time slot as follows  
 $T_{header\ maximum} = 14 * T_{header\ Nominal}$   
 $T_{response\ maximum} = 1.4 * T_{response\ Nominal}$   
 $T_{frame\ maximum} = T_{header\ maximum} + T_{response\ maximum}$

#### IV. Result and Discussion

The calculations of real-time properties performed for each case study is limited to the specific transfer of LIN messages on the bus. This is due to the time variance different implementations of nodes will have. Various types of internal architecture of microprocessors and FPGAs etc. will affect the delay between when an event occurs or a timer time out and the time an updated message leaves the slave.

The Lin maximum frame size is 91 bits when we consider when the data field is 2-bit header is 34 bit and checksum is 10 bit. The latency for local interconnect network using electrical wire and optical fiber may calculated on the basis of above formulations.

**Table 1 The differences in message transfer times and frame delays for different sized messages**

Frame Size	Time Base = 5 ms						Time Base = 5 μs					
	9.6 kbps			19.6 kbps			9.6 mbps			19.6 mbps		
	FD (ms)	MTT (ms)	Slack (ms)	FD (ms)	MTT (ms)	Slack (ms)	FD (μs)	MTT (μs)	Slack (μs)	FD (μs)	MTT (μs)	Slack (μs)
1	10 = t1	8	2	5 = t1	4	1	10 = t1	8	2	5 = t1	4	1
2	10 = t2	9.5	0.5	5 = t2	4.7	0.3	10 = t2	9.5	0.5	5 = t2	4.7	0.3
3	15 = t3	10.9	4.1	10 = t3	5.5	4.5	15 = t3	10.9	4.1	10 = t3	5.5	4.5
4	15 = t4	12.4	2.6	10 = t4	6.2	3.8	15 = t4	12.4	2.6	10 = t4	6.2	3.8
5	15 = t5	13.9	1.1	10 = t5	6.9	3.1	15 = t5	13.9	1.1	10 = t5	6.9	3.1
6	20 = t6	15.3	4.7	10 = t6	7.7	2.3	20 = t6	15.3	4.7	10 = t6	7.7	2.3
7	20 = t7	16.8	3.2	10 = t7	8.4	1.6	20 = t7	16.8	3.2	10 = t7	8.4	1.6
8	20 = t8	18.2	1.8	10 = t8	9.1	0.9	20 = t8	18.2	1.8	10 = t8	9.1	0.9
Sleep	20 = tsleep	18.2	1.8	10 = tsleep	9.1	0.9	20 = tsleep	18.2	1.8	10 = tsleep	9.1	0.9
wake-up	twake-up	1.25	1.8	twake-up	0.6		twake-up	1.25	1.8	twake-up	0.6	
	Max Difference		4.7	Max Difference		4.5	Max Difference		4.7	Max Difference		4.5

When looking at response times, it is the worst case that is of interest. When looking at a schedule for a LIN-network, the frame-delay (time slot) for each message gives a bit of slack before the next message is sent as seen in Figure 8. This implies that, for example a slave node receives a complete message before the end of the time slot. Table 1 shows the differences in message transfer times and frame delays for different sized messages

To be noted is that the slack for a three-byte message when running a network at 19.2 kbps is quite substantial. In this case a time base of 6 ms would be better. The difference between frame delays and transfer times for the three case studies in this chapter will however be of maximum 6% and is therefore negligible

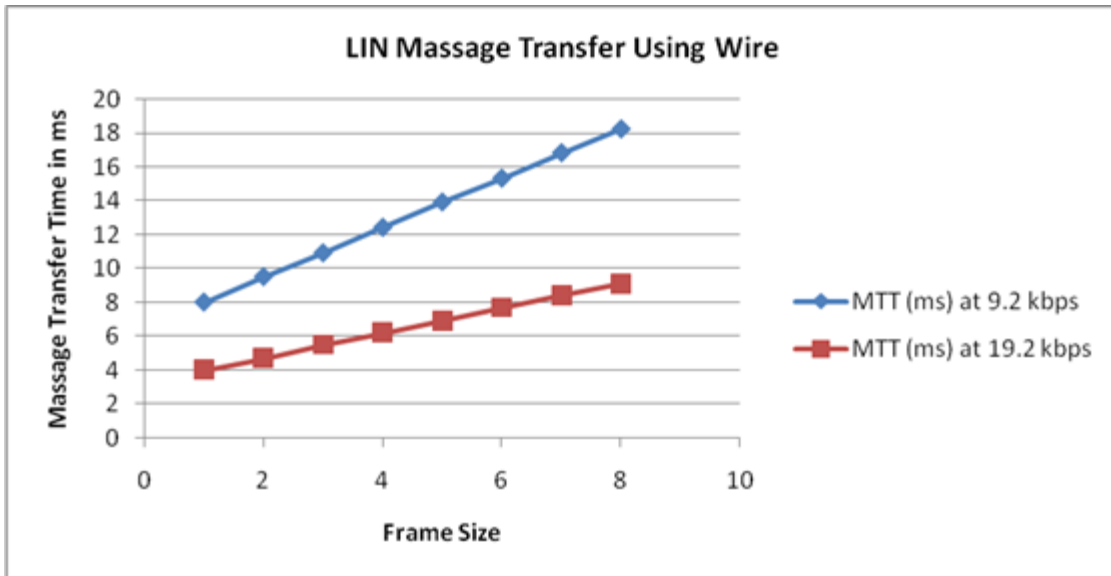


Figure 8: LIN Message Transfer Time using wire at 9.2 & 19.2 kbps

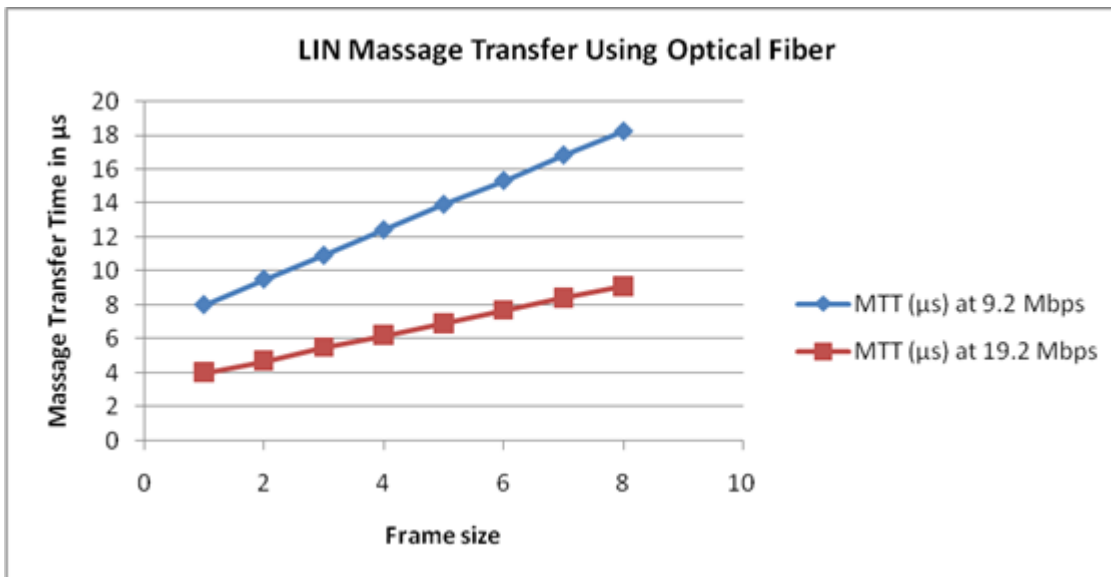


Figure 9: LIN Message Transfer Time using optical fiber at 9.2 & 19.2 Mbps

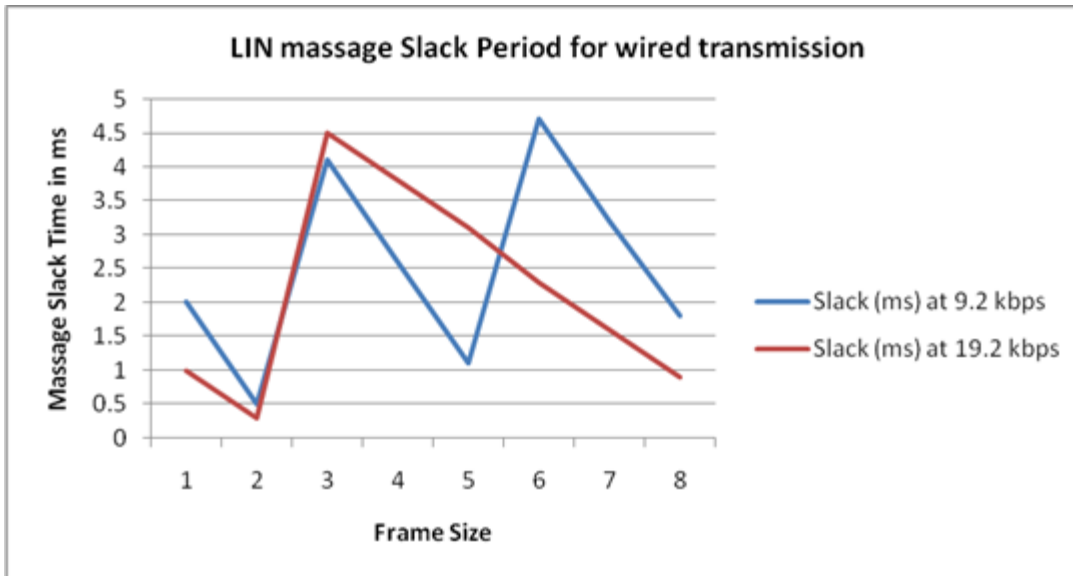


Figure 10: LIN message stack period for wired transmission at 9.2 & 19.2 kbps

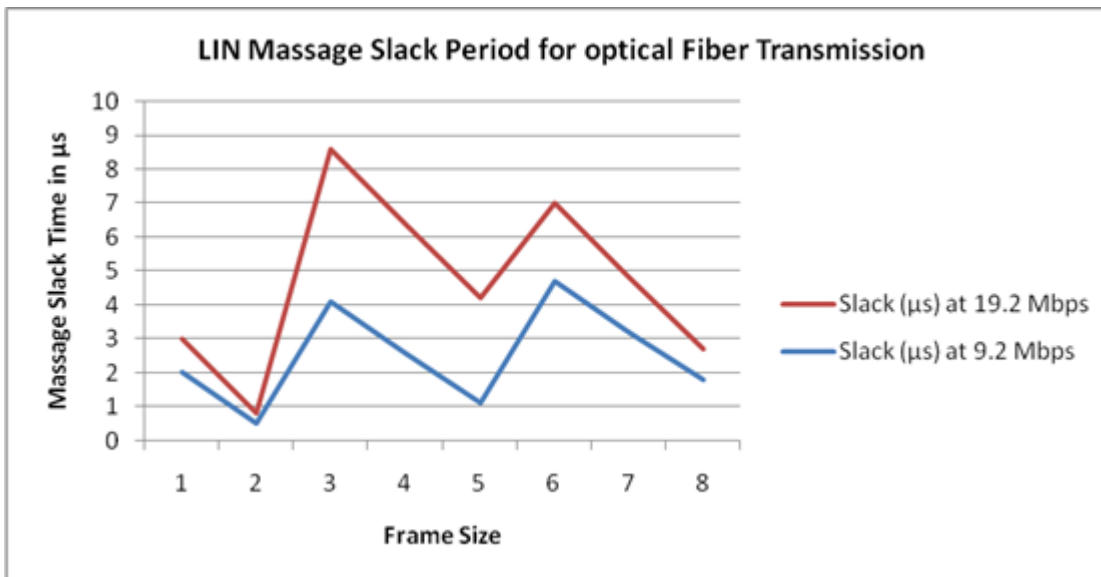


Figure 11: LIN message stack period for optical fiber transmission at 9.2 & 19.2 Mbps

The comparison of transmission latency for wire and optical fiber is shown in above figures with the help of message transfer time and slack time. Figure 8 shows message transfer time at 9.2 kbps and 19.2 kbps with respect to frame size. As the message frame size is goes on increase the message transfer time is also goes on increase, which increase in signal latency. In case of 9.6 kbps transmission speed with time base 5 ms the frame delay (FD) observed from 10 ms to 20 ms and message transfer time is 8 ms to 18.2 ms. The slack period is from 0.5 ms to 4.7 ms. The observed time period for signal latency is very high, therefore to reduce the signal latency on wire the transmission speed is increased to 19.6 kbps. The resulted observations for frame delay is from 5 ms to 10 ms. The message transfer time is from 4 ms to 9.1 ms. The slack period is varied from 0.3 ms to 4.5 ms. The signal latency by means of MTT and slack period is improved by 50 % as shown in figure 10 and 11. In our designed case the targeted speed is in mbps and the required time period is in micro second. To achieve the aim with low cost designed considerations are made using plastic optical fiber. The observations made at 9.6 mbps and 19.6 mbps with time base of 5 μs. At 9.6 mbps the frame delay is varied from 10 μs to 20 μs, message transfer time is from 8 μs to 18.2 μs and slack time is 0.5 μs to 4.7 μs. To get more improvement in signal latency using optical fiber computation are observed at transmission speed of 19.6 mbps. This shows much more improved results, as



frame delay noted from 5  $\mu$ s to 10  $\mu$ s. Message transfer time 4  $\mu$ s to 9.1  $\mu$ s therefore the slack period reduced to 0.3  $\mu$ s to 4.5  $\mu$ s. The comparisons for MTT and slack time using optical fiber is shown in figure 3 and 4. The above compared results show that the signal latency can be improved with help of plastic optical fiber. As the signal time period is reduced from milliseconds to microseconds. Used optical fiber reduced EME value and 3, as transferred signals with the help of light. EME values low, the slew rate of the signals and also the bandwidth of the data transferred is also controlled.

## References

- [1]. R. J-Michel, M. Steyaert "An EMI Resisting LIN Driver in 0.35-micron High-Voltage CMOS" IEEE Journal of Solid-State Circuits, vol. 42-7 pp. 1574-1685, July 2007.
- [2]. J.G. Janschitz et al. "An EMI robust LIN driver with low electromagnetic emission", Electromagnetic Compatibility of Integrated Circuits (EMC Compo) 2015 10th International Workshop, 2015, pp. 2145-2153.
- [3]. J. P. Dsouz, R. Reddy, S. Yellampalli, "Diagnostic protocol stack on LIN interface", IJERT, Vol. 3-11, pp 1208-1211, 2014.
- [4]. W. J. Dally, "Performance analysis K-ary n-cube Interconnection networks", IEEE transaction on computer, Vol. 39-6, pp. 775-785, 1990.
- [5]. C. P. Kruskal, M. Snir, A. Weiss, "The distribution of waiting times in clocked multistage interconnection network", IEEE transaction on computers", Vol. 37-11, pp 1337-1352, 1988.
- [6]. J. P. Patel "Performance of processor memory interconnections for multiprocessors", IEEE transaction of comters, C-30-10, pp 771-780, 1981.
- [7]. A. C. Stanca , F. Sandu, " Solution to improve the features of local interconnected networks" Bulletin of Transilvania university engineering sciences, vol. 6-55 (2), pp. 97-105, 2013.
- [8]. R. Hegde, S. Kumar, K. S. Gurumurthy, "The impact of network topologies on performance of the in-vehicle network", International journal of computer theory and engineering, Vol. 5-3, pp. 405-410, 2013.
- [9]. C. Gabiel, H. Horia, "Integrating sensor devices in LIN bus network", 26<sup>th</sup> International spring seminar on electronics technology, pp. 150-153, 2003.
- [10]. Y. Sun, "Optimal scheduling algorithm in vehicular real time system" Journal of Network, Vol. 9-3, pp. 785-792, 2014.
- [11]. G. Lipari, " Efficient scheduling of real time multitask application in dynamic system", IEEE 33<sup>rd</sup> conference, pp. 1530-1544, 2000.
- [12]. T. Simunic, J. W. Rozenblit, "Reduction of signal delay and cross-talk in electronics packages using expert system technique", Proceeding of IEEE electrical performance of electronics packaging, pp. 61-64, 1993.
- [13]. A. Biberman, K. Bergma, "Optical interconnection for high performance computing system ", Journal of Reports on progress in Physics, Vol. 75, pp. 1-15, 2012.
- [14]. R. Chauhan, R. A. Pagare, " Review of drivers based optical fiber communication using optical USB cable", Internatonal journal of advancements in research and technology, Vol. 4-8, pp-49-53, 2015.
- [15]. S. Marikura, K. Kinoshita, K. Numata, S. Furusawa, "High speed POF transmission technology and its standardization ", Proceeding of 27<sup>th</sup> European conference on optical communication " Vol. 6, pp. 242-248, 2001