

LIN Protocol-Emerging Trend in Automotive Electronics

Sandeep R. Gadekar¹ and Shilpa Kodgire²

*¹Electronics & Communication Dept.
MIT A'BAD, MH*

*²(Electronics & Communication Dept.)
MIT A'BAD, MH.*

Abstract

This is an era of automation where replacement of manual efforts are achieved by mechanical power in all degree of automation. Automotive motorized window or door apparatus which closes automatically involve risks for trapping, squeezing or injury to people. They must reverse their movement in case the force applied by the motor gets higher than one normalized limit. In this paper we describe Anti-Pinch Control System for Electrical Window as application of LIN protocol. This paper describes the possibility and advantages of using LIN as multiplexed electrical system in vehicle. LIN is communication protocol designed for controlling simpler electrical components in a vehicle like sensors, switches etc.

1. Introduction

Today, electronic components and systems account for over 20% of the cost of a high end passenger car. In-vehicle networks are already successfully applied to body and power train control systems. As far as in-vehicle control networking technology is concerned, an emerging growth area is in the adoption of a low cost body control sub-bus that complements CAN.

Many body control functions are often include simple operations such as activating lights, wipers, windows etc. These are considered soft requirement real time systems that do not necessarily need the response that can be provided by CAN. Additionally, there is also the desire in the industry to integrate a network node into a sensor or actuator. LIN is currently the candidate technology for these applications since it is a complementary technology to CAN but at a much lower cost.[4]

The automotive power-operated windows are governed by international standards, like MVSS118 in USA or 74/60/EEC in Europe. For those relatives to the elimination of danger to children, the requirements can be summarized as follows,

- Detection area: 4 mm to 200 mm,
- Maximum pinch force = 100N,
- Reverse direction on a pinch.

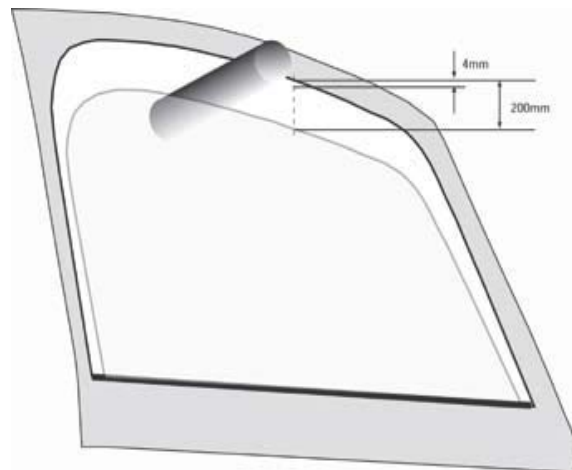


Figure 1: Anti-pinch Window Lift Rules.

Different detection strategies are possible to determine whether an obstacle is entering the critical pinching area:

1. Without mechanical contact: It reacts before the pinch effort is exerted on the object. This is the optimal protection since no force is applied to the obstacle. It is also independent to vibration, aerodynamic variations, deformations. But, it requires integrated sensors (infrared, ultrasonic, etc) with their electronic modules and wires leading to additional costs.
2. With mechanical contact: The pressure measurement will tell the system an object is being pinched. Also there are two fundamental technologies available:

Direct measurements: Force sensors or contactors are integrated on the door seal. These solutions are inherently high cost and reduce the styling for window/door designs.

Indirect measurements via physical monitoring. This is a globally cost optimized solution.[3]

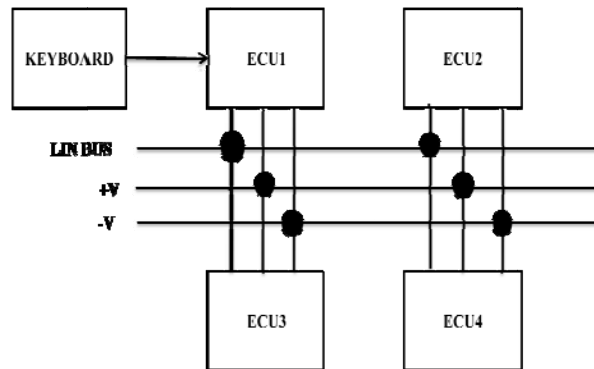


Figure 2: Block Diagram Representation.

The Figure 2 above shows the basic block diagrammatic representation. This is basically a LIN network which consist of four Electronic Control Unit(ECU) and keyboard.

The LIN network is basically divided into two units:

- 1.One Master Control Unit.
- 2.Three Slave Unit.

Master Control Unit:

The Master control unit consists of ECU1 which initiates all the communication. It performs the slave communication task that includes the transmit and receive task. In addition, the master node performs the master transmit task. The master task decides when and which frame will be transferred on the bus. In this way there is no bus arbitration. It handles all the operation of networking and communication with other peripherals. This ECU1 take the input from the keyboard and controls remaining three slave units i.e ECU2, ECU3, ECU4. It handles data transmission and reception with them as well as it controls window lift operation of front seat right hand side window i.e driver side window.

Slave Control unit:

The Slave control unit consist of ECU2, ECU3, ECU4. They performs the slave communication task that includes the transmit and receive task.

ECU2: It is an front seat left side window control unit which shift the position of the window as per the response from ECU1.

ECU3: It is an back seat right side window control unit which shift the position of the window as per the response from ECU1.

ECU4: It is an back seat left side window control unit which shift the position of the window as per the response from ECU1.

2. Why LIN?

There are four distinct areas of applications using network communication and protocols for controlling

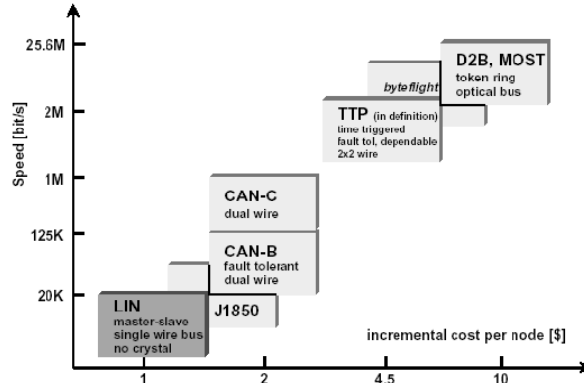


Figure 3: Data Rate versus Cost for Automotive Real time Systems.

different functions in vehicles today. Each of these areas needs its own type of protocol and there is no single "universal" protocol that can be used as a solution for all types. Each type requires specific features:

Conventional body and powertrain applications, use protocols with known real-time properties, mainly CAN. Multimedia applications, calling for protocols that should provide high bandwidth and speed and even wireless interconnection. Bluetooth, MOST or Firewire are typical examples. Safety critical applications, needing protocols that are fault tolerant and reliable. X-by-wire is an emerging market that calls for protocols like TTP/C (Time- Triggered Protocol)FlexRay and TT-CAN (Time Triggered CAN). Mechatronic type applications such as smart sensors and actuators, or even complex ECUs with simple communication needs. These applications are addressed by protocols like LIN, TTP/A and other OEM (Original Equipment Manufacturer) specific protocols. Figure 3 shows how the different protocols are related to each other. CAN protocol is adopted amongst most of the vehicle manufacturers and in many other areas as well.

3. Performance and resource

Requirements of LIN versus CAN

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

Table 1: Comparison of general features of LIN and CAN[2]

Feature	LIN	CAN
Media Access Control	Single Master	Multiple Masters
Typical Bus Speed	2.4, 9.6 and 19.2 kbps	62.5...1000kbps
Multicast Message Routing	6 bit identifier	11/29 bit identifier
Typical Size of network	2...16 nodes	4...20 nodes
Data Byte per Frame	0...8	2...8
Transmission time for 4 data bytes	6 ms at 19.2 kbps	0.8 ms at 125 kbps
Error detection (data field)	8-bit checksum	15 bit CRC
Physical Layer	Single wire, 40V	Shielded, Twisted pair, 5V
Quartz/ ceramic resonator	Master only	All nodes
Relative Cost per Network connection	0.5	1.0

The main economical factors in favor of LIN are the single-wire implementation, the low cost of silicon implementation since the communication is based on UART/SCI-interface and the avoidance of high cost quartz or ceramic resonators in slave nodes. However the drawbacks are lower bandwidth and less effective bus access scheme with the master- slave configuration. The main features of the LIN and CAN protocol can be viewed in Table 1 above.

4. Pinch Detection Algorithm

Usual pinch-detection algorithm operation is using indirect measurements out of the window-lift system:

- Current (torque)
- Position (speed)

The algorithm aggregates two techniques based on :

-Calibrated torque stored in non-volatile memory: Preliminary learning sequence is executed and torque values are stored in memory. This is quite memory consuming and requires regular calibration sequences.

-Speed derivate calculation: Interesting method since it requires less memory but needs more computing power.

Pinching Condition:

A pinch is detected by comparing the current measurement with a reference,

$$M(t) \text{) } k_j \times i(t) = K \times i(t)$$

The threshold can be determined out of the Motor Torque constant (K) combined with the response time of the system.

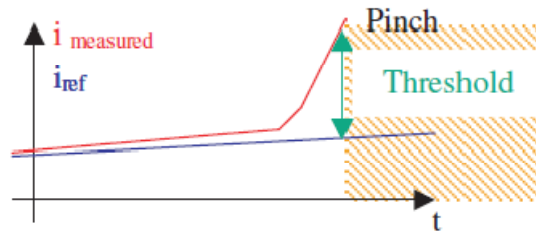


Figure 4: Condition for Pinch.

One solution to eliminate need for calibration is to calculate expected current all along the movement.

One pinch condition makes the speed decreasing and the current growing up associated with increasing torque. Those two conditions are used to determinate the occurrence of a pinch. Speed derivate is preferred for its higher robustness to noise or fast disturbances.

As shown in Figure 5 The Current Reference (i_{ref}) is permanently calculated until the speed increases or remains unchanged. As soon as the speed decreases rapidly (Derivate becomes negative), i_{ref} is frozen. Current through the motor (i_{mot}) continues to be measured and compared with Current Reference + Margin. In case i_{mot} gets higher than the limit, then a Pinch is declared and several actions are taken First is to stop the motor and reverse its rotation to release the obstacle.[3]

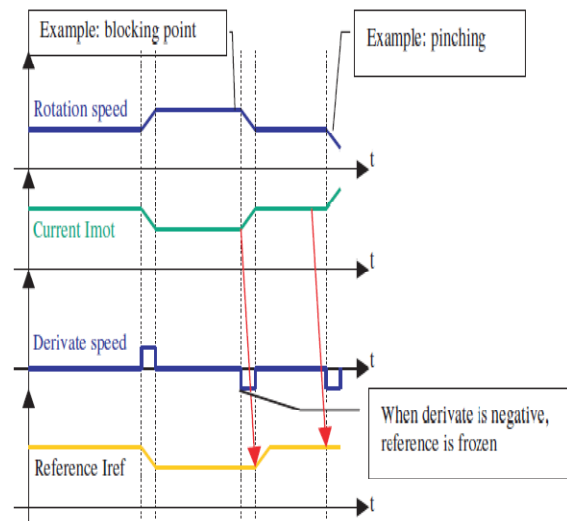


Figure 5: Pinch Detection

5. Hardware and Software

This section describes the hardware required for actual implementation and the software used for designing and code compilation.

5.1. Hardware Description(μ C & Transceiver):

CAN Controller 90CAN128 is used to initialize all task. The voltage level of the LIN(8V to 40V) is not compatible with the TTL voltage level so there is need of transceiver, so we have selected the LIN transceiver TPIC1021 which converts the LIN voltage level to TTL compatible voltage level.

5.2. Software Description(Tools used):

The software tools used for designing and compilation of the codes were OrCAD v9 and AVR Studio v6.OrCAD is used for schematic preparation along with OrCAD PSpice A/D for circuit simulation. AVR Studio is an Integrated Development Environment (IDE) for writing and debugging AVR applications.

6. Conclusion:

LIN Bus Architecture will be a significant factor in reducing cost of non-mission critical actuations and controls. LIN is not a full replacement of the CAN bus. But the LIN bus is a good alternative wherever costs are essential and speed/bandwidth is not essential.

References

- [1] Xiao Zhaoxia; Li Youcheng "Hardware Design of Automobile Door with Local Interconnect Network Bus "Control, Automation and System Engg.(CASE), International conference on Communication, Computing & Processing, Signal processing & Analysis 10.1109/ICCASE.2011
- [2] J. Will Specks, Antal Rajnak, *LIN Protocol, development tools and software interfaces for Local Interconnect Networks in vehicles*, 9th International Conference on Electronic systems in vehicles, Baden-Baden October 5 2000
- [3] Yi Wang "A Low Cost Anti Pinch System for Electrical Window" Information and Computing (ICIC),2011 Fourth International Conference on Computing & Processing (Hardware/Software).
- [4] Wang Yi; Qiu Yunfeng; Yin Jie "Application of Automobile Power Window for Peripheral Interface Controller" Electronics Measurement & Instruments (ICEMI) 2011 10th International Conference on Communication, Networking & broadcasting.

