Stanford University Libraries 1
36105110419897
TL
272.53
.A38
2001
ENG

S. Krueger
W. Gessner
(Eds.)

Innovation

VDI|VDE-IT

Advanced Microsystems for Automotive Applications 2001

Springer

BMW EXHIBIT 1011

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milano
Paris
Singapore
Tokyo

Engineering ONLINE LIBRARY
http://www.springer.de/engine/

Sven Krueger · Wolfgang Gessner (Eds.)

Advanced Microsystems for Automotive Applications 2001

With 193 Figures

Springer

SVEN KRUEGER
VDI/VDE-Technologiezentrum Informationstechnik GmbH
Rheinstr. 10B
D-14513 Teltow
e-mail: krueger@vdivde-it.de

WOLFGANG GESSNER
VDI/VDE-Technologiezentrum Informationstechnik GmbH
Rheinstr. 10B
D-14513 Teltow
e-mail: gessner@vdivde-it.de

ISBN 3-540-41809-1 Springer-Verlag Berlin Heidelberg NewYork

Cip data applied for

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in other ways, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution act under German Copyright Law.

Springer-Verlag is a company in the BertelsmannSpringer publishing group
http://www.springer.de
© Springer-Verlag Berlin Heidelberg New York 2001
Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera ready by authors
Cover-Design: de´blik, Berlin
Printed on acid free paper SPIN: 10796954 68/3020/kk - 5 4 3 2 1 0

Preface

In the year 1980 the value of electronics in the automobile was 2%. It grew up to 17% in the year 2000, in some vehicles it reached even 30%. Experts expect the car electronics sector to grow in the next years with an average annual rate of 8,5% (Automotive World, Dec. 2000). With the increasing use of electronics also the application of microsystems – the invisible pacemakers behind visible new features of comfort, safety and performance – steadily increased from that time on, when the first pressure sensor was applied in an engine management system. Since then enormous progresses have been made. Today most of the innovations in the car are determined by electronics and in most of the electronic systems microsystems or microsystem components play the decisive role: pressure measurement systems, mass flow sensors, accelerometers and angular rate sensors have become state-of-the-art. A modern premium class car contains more than 100 sensors of which at least 40 are part of a microsystem – systemically integrated and equipped with intelligence.

Looking back only some years to the first AMAA publication in 1998 or to the first beginning of the AMAA initiative in 1995 we easily notice, that a series of microsystems applications - once a vision - in the meantime became reality in the car. Others already existing could be drastically improved. High growth rates could be realised – particularly for microsystems in engines - and enormous cost reductions were achieved by introducing new fabrication technologies, which allowed to meet the customers' price-performance requirements better, and thus made the breakthrough of new systems possible. Nevertheless a series of items remain on the agenda - e.g. microsystems for the improvement of safety, of engine management efficiency and of comfort - and some of them even gained a higher importance. More severe environmental and lifetime requirements as well as increased safety regulations (as FMVSS208) set new challenges to be faced and under technological aspects also – with a mid-term perspective - the forthcoming 42V system and changing EE design, multimedia applications and new human-machine interface concepts.

The publication in hand is the fourth of a series. It gives an overview on the presentations given at the 5th International Conference on Advanced Microsystems for Automotive Applications. It is a cut-out of new priority topics in the area of automotive applications of microsystems showing mid-term perspectives from a 2001-point-of-view. The resonance of the companies this year concentrated on interfaces, standards and networks. Microsystems developments in view of future multimedia applications in vehicles is a further central issue as well as key

applications based on new sensor developments and sensor fusion systems. As in previous years safety applications are a topic with high priority.

The AMAA certainly is much more than this book is able to express. It has become over the years a well established forum where global suppliers, car manufacturers and SMEs present visions, requirements and solutions. The event is successful, because its format is a platform for the exchange of new ideas – building and interlinking networks, and connecting the different views of technologists and users. This is useful in itself, but in time it also generates specific technology transfer and co-operation among industrial partners.

My explicit thanks are addressed to Mr. Hernandez-Ros of the European Commission for the financial support of the event through the Innovation Relay Centre Northern Germany. I would also like to thank the members of the AMAA Honorary Committee who accompanied the initiative over many years with their trusty advice and the Steering Committee – the real backbone of the AMAA since its first beginning - for their assistance in selecting the contributions. Particular thanks go to Roland Müller-Fiedler, Robert Bosch GmbH, John P. Schuster, Motorola, and Bob Sulouff, Analog Devices, and the colleagues of DaimlerChrysler for their engagement in the this-year's event and for the stimulating conversations we had. I thank the authors for their excellent papers and for their co-operation in realising this publication, and all individuals and companies without whose support neither this book nor the event would never have been realised. Sincere thanks are addressed to the Innovation Relay Centre team and Sven Krüger who was responsible for the organisation of the AMAA 2001.

Teltow/Berlin, May 2001

Wolfgang Gessner

Table of Contents

Application Opportunities of MEMS/MST in the Automotive Market: The Great Migration from Electromechanical and Discrete Solutions 1
R. H. Grace, Roger Grace Associates

Using Micro Devices in Automotive Applications 17
M. Schmidt, DaimlerChrysler AG

Session 1: Multimedia

Passenger Information and Entertainment – Multimedia Trends and Architectures 23
M. W. Schneider, Delphi Delco Electronic Systems

The Automotive Internet – An Advanced Automotive Application 37
P. Rößger, CAA AG

Organic LED (OLED) for Automotive Display Applications 47
F. Bouckaert, Pioneer Technology

Session 2: Interfaces – Communication for Microsystems

Intelligent Tyre Technology 63
J. Hakanen, Nokian Tyres plc

Wireless Tire Sensors Based on Amorphous Magneto-Elastic Materials 83
M. Löhndorf, CAESAR

14V / 42V Power Line Communication for Automotive 89
Y. Maryanka, Yamar Electronics Ltd.

Transceivers for Optical Networks in Automotive Applications 95
J. Wittl, Infineon Technologies AG

Development of Smart Initiators 109
B. Rainwater, Philips Semiconductors

New Architectures for Faster Automotive Design Cycles 121
G. Teepe, Motorola GmbH

Session 3: Media Sensors / Severe Conditions

Pressure Sensors for Automotive Applications – with New One Chip EPROM Technology 131
T. Sakai, Fuji Electric Co. Ltd.

A Mass Flow Sensor for Fuel Injection Quantity Measurements in DI-Systems 139
U. Schmid, EADS Deutschland GmbH

A Novel Multifunctional Oil Condition Sensor 157
M. Scherer, Robert Bosch GmbH

Air Quality Sensor for Air Condition Units 167
K. D. Frers, paragon AG

Session 4: Safety Applications

Accelerometers Utilising MEMS Technology and their Design Considerations for Automotive Restraint Systems 177
S. Naidu, Visteon Corporation

Design and Performance of the SAR10 Rate Gyro 189
T. Kvisterøy, SensoNor asa

Micromachined Sensors for Advanced Microsystems 201
B. Sulouff, Analog Devices Inc.

High Reliability and Low Cost Uncooled Microbolometer IR Focal Plane Array Technology for Commercial Application 209
J.-L. Tissot, CEA/LETI-DOPT

Poster Presentations

Integrated Electronics for Bus Systems Igniters 221
T. Görnig, TEMIC TELEFUNKEN microelectronic GmbH

Advanced Etch Tool for High Etch Rate Deep Reactive Ion Etching in Silicon Micromachining Production Environment 229
A. Schilp, Robert Bosch GmbH

Micro-Electro-Mechanical Acceleration-Sensitive Switch 237
T. Frank, Little Things Factory GmbH

Tyre Pressure Monitoring Microsystems 245
R. Grelland, SensoNor asa

High Temperature Pressure Sensors – Potentials for New Concepts in Automotive Applications 253
F. Solzbacher, First Sensor Technology GmbH

Virtual Sensors for Vehicle Dynamics Applications 267
U. Forssell, NIRA Dynamics AB

Building Automotive LIN Applications 279
H.-C. von der Wense, Motorola GmbH

Low Cost 360° Tilt Compensated Electronic Compass 293
H. Grüger, Fraunhofer Institut für mikroelektronische Schaltungen und Systeme

List of Contact Addresses 299

List of Keywords 303

Building Automotive LIN Applications

H.-C. von der Wense, A. J. Pohlmeyer*

Motorola GmbH
Schatzbogen 7
81829 Munich, Germany
Phone: +49/89/92103-882, Fax: +49/89/92103-820
Email: H.Wense@Motorola.com

Motorola, Inc
41700 Six Mile Rd
Northville, MI 48167-3479, USA

Phone: +1/248/347-7926, Fax: +1/248/347-6262
Email: A.J.Pohlmeyer@Motorola.com

Keywords: communication protocol, distributed system, LIN, mechatronics multiplex Bus-architecture

Abstract

As automotive electronic systems become more and more spread, the necessity for a complete portfolio of communication standards for automotive applications becomes obvious. In the low cost Class A Smart Sensor & Actuator arena Local Interconnect Network (LIN) is the top candidate to be used as a multiplex standard all over the world.

The LIN concept includes not only the physical layer and the data link layer; it also defines the tool interfaces and network description to speed the development process for in vehicle networks.

This article describes the development process for LIN applications and gives an example of a typical LIN application, explains the benefits of this solution from the major aspects, the usage of the different tool interfaces, and its representation on the data link layer and the physical layer.

1 Introduction

In the last few years following trend in automotive electronic design could be noticed: More and more functions have been put into the car and more and more of these functions are enabled by local intelligence. The key for those distributed systems is multiplex networking.

Several multiplex protocols have established in the car and each of the protocols has its specific domain. Most popular of these protocols is CAN with its domains as engine control network and main control network in the body control area. MOST[1], D2B and FireWire[2] are the top candidates in the area of Infotainment. Time Triggered Protocols, such as TTP[3], Flexray[4], and Byteflight[5] are the preferred communication carriers for safety control systems, such as braking and steering. In the low end communication area LIN[6] has become the top candidate. Typical applications for LIN are in the area of cost critical distributed body electronics where the performance and versatility of CAN is not required.

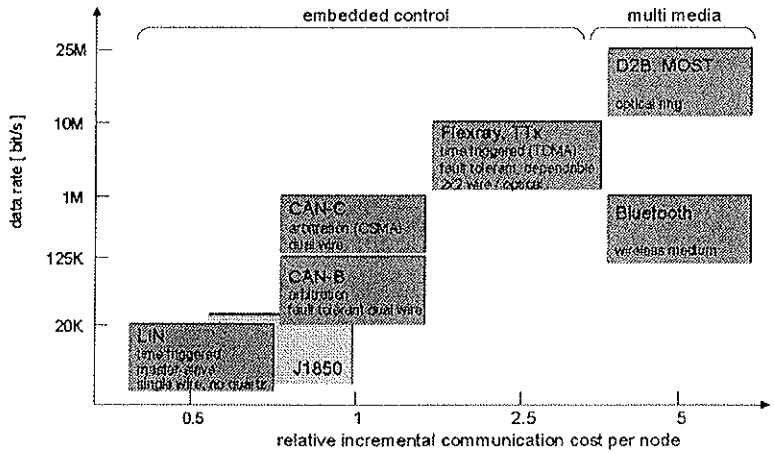

Fig. 1. Automotive network systems[7]

LIN has been created in the growing necessity for a standard in the low-end communication area. Several car-makers had their own proprietary communication running in their distributed networks, sometimes the solution for low end communication could differ already from department to department, resulting in several non compliant subnets in one car. In 2000 the LIN-Consortium has been founded to address the missing standard.

LIN is the first communication system that specifies from the beginning not only the protocol properties but also tool interfaces: Interfaces to network configuration and network development tools are well defined. Also the interface of the

networking portion to the application has been specified. Tools are available to ease the development of LIN systems. The development of single nodes of a LIN system is possible when the configuration of the network is known. (This has been proved in a case study at one of the founding members – The network configuration has been defined at the car manufacturer. The nodes have been designed at the different suppliers with the prior knowledge of the network configuration and the application requirements. When the nodes were attached to the network, the system was immediately functional.).

2 LIN Basics

LIN is a low-cost single wire network. The physical layer is ISO 9141 compliant. In order to meet EMC requirements the slew rates are controlled. The protocol is a simple master slave protocol based on the common UART format. In order to enable communication between nodes clocked by cost effective RC-Oscillators, synchronisation information is transmitted on the bus, with which slave nodes can synchronise themselves with the master clock, which can be regarded as accurate. The speed of the LIN network is up to 20 kbit/s and the transmission is protected by a checksum.

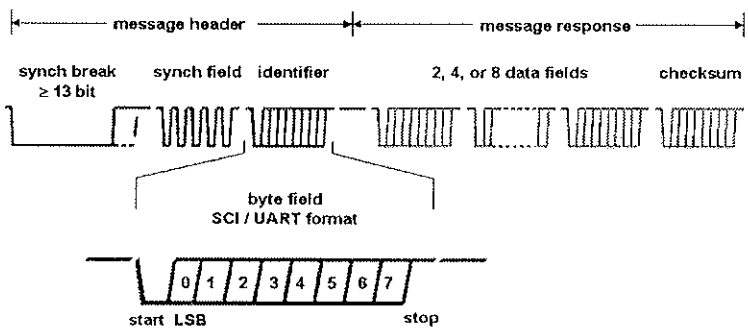

Fig. 2. LIN Message Frame

The LIN Protocol is message-identifier based. That means that the identifiers do not address nodes directly but denote the meaning of the messages (similar to CAN). This way any message can have multiple destinations (multicasting). The master sends out the message header consisting of a Synchronisation Break (serving as a unique identifier for the beginning of the frame), a synchronisation field carrying the clock information, and the message identifier, which denotes the meaning of the message.

Upon reception of the message identifier the nodes on the network know exactly what to do with the message. One of the nodes sends out the message response and the others either listen or do not care. Messages from the master to the slave(s) are carried out in the same manner – in this case the master also sends the response.

LIN messages are scheduled in a time triggered manner, this allows predictability and the accurate calculation of latency times. Since the master sends out the headers, it is in complete control of the scheduling and is also able to change the table according to the requirements of the applications running in the sub-system.

The complete LIN Specification defines the Protocol and the physical layer as well as the interfaces to tools and application programs. This way it has been possible to use LIN from the beginning in a very comfortable way.

3 Development of a LIN Network

LIN has been developed to serve as local subnet to networks with higher performance such as CAN and thus replace hard wiring. Signals are required to be transported within a sub-domain as well as from one subnet to other subnets (e.g. door to door). The higher performance network serves as backbone.

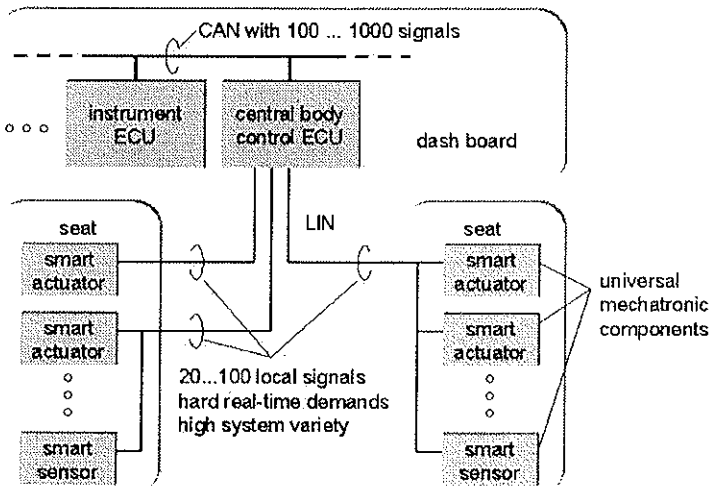

Fig. 3. LIN as subnet of CAN

As part of complex vehicle networks LIN systems cannot be developed independently. Applications require signals that are transported from other parts of the vehicle. The number of signals and their properties such as real time requirements and the speed of the different bus systems require the analysis of the complete network architecture available.

The optimisation of the network requires the consideration of criteria such as:

- number of signal to be communicated;
- signal latencies and other real-time requirements of functions;
- speed, bandwidth, and medium of the bus;
- electro-magnetic compatibility (EMC);
- fault tolerance or fail safety;
- cost per electrical interconnection versus cost of local intelligence;
- ECU variety cost by customer options and over vehicle platforms, system scalability;
- cost and reliability of systems integration in silicon and mechatronics;
- availability of tools and software;
- existing infrastructures and development skills, architecture and networking legacies.

The network architecture depends on the weight and balance of the above items. The partitioning of the network is beyond the scope of this paper – we will concentrate on the aspects of LIN. The process for network development in LIN starts with the definition of the network objects and their properties.

4 The Door Example

One of the popular domains of LIN is body control. In our example we have the task to develop the complete door subsystem that is applicable as driver door as well as passenger door. It should feature window lift, door locks, and mirror with their basic functions. There should also be the possibility for optional functions in the mirror, such as electrochrome, temperature sensor. Different keyboards have to be supported, in order to have optional back door support as well as optional seat positioning support.

Our door consists of a LIN Master ECU operating as a gateway between the LIN-network and the main network of the car and several slave nodes that represent the mechanic functions of the door. These slave nodes include window-lift, door-lock, control panel, and mirror. Traditional approaches are using one central control unit in the door with a high amount of wiring to the different actuators. The mechatronic approach combines the ECU with the mechanic and thus creates a distributed subsystem in the car. Figure 4 shows such a door setup with Motorola mechatronic components. These components are connected with each other via the power supply and the single wire LIN.

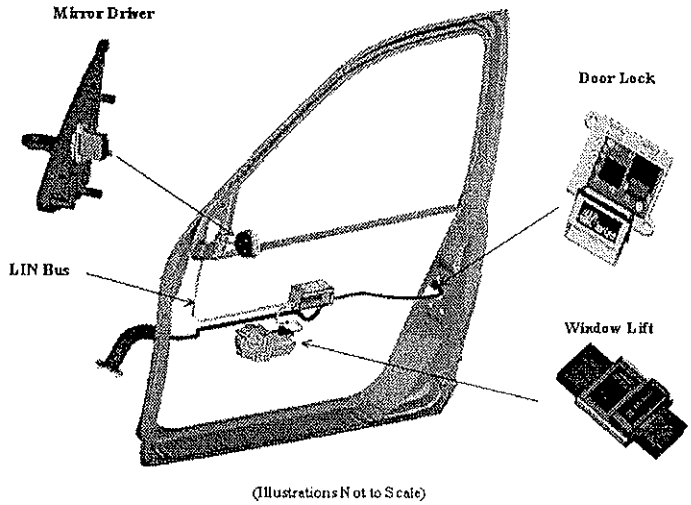

Fig. 4. Mechatronic Door and its components - based on LIN

Components of the Door and their Functions

Our door subsystem contains a keyboard, a mirror, a window lift, and a door-lock. On the driver's side the keyboard features the controls for mirrors, window-lifts / window-lock, and central lock. On the passenger side only the window control is available. Optional there could be the control of the seat position.

The window lift offers the following functions: window up / window down, express up / express down, anti-pinch detection.

The door latch can be locked/unlocked and returns its status.

The mirror is the most complex part of this door: it incorporates not only the position of the mirror, it also features mirror heating, electrochrome and a light (e.g. as direction signal/flasher/puddle light).

The gateway from/to the main body network is incorporated in the keyboard controller. This reduces latency times when e.g. the driver is controlling actuators in the passenger doors (e.g. mirror, window). These control signals have to travel via the main body network to the subnet on the far side of the car.

- Keyboard – driver's door
- Gateway
- Control for 2 mirrors
- Control for 2 front door windows
- Optional:
- Control for 2 back door windows

- Seat control
- Window lift
- Window up/down
- Express up/down
- Anti pinch control
- Door latch
- Lock/unlock
- Door latch status (open/closed)
- Lock status (locked/unlocked)
- Mirror
- Mirror position (x/y)
- Mirror heating
- Fold
- Optional:
- Electrochrome
- Outside temperature sensor
- Flasher/puddle light

The above list shows already that the door has already become a fairly complex system. From the above functions the necessary signals for the door can be derived. We do not discuss all signals in detail, as it would exceed the time available. We just pick some interesting signals and investigate their properties.

5 Signals

The ECUs of the door communicate with signals. The signals represent the electrical properties of the ECUs. For the mirror these could be x_pos and y_pos as position status of the mirror. In the analogue world they would be represented e.g. by voltage. In the digital world numbers replace them. Signals have their specific properties, such as source, destination(s), resolution, and time deadlines. Their contents represents analogue (e.g. position or temperature) or digital (on/off) values. The nodes sending out signals "publish" them, the signals which are received and used in the application are "subscribed".

Multicast Messaging

Some of the signals have multiple destinations. E.g. the Door_lock-signal is subscribed by the latch and also by the other door systems. LIN uses Message Identifiers rather than node identifiers. This bears the possibility that a single message can be received by more than one node. These nodes receive e.g. the message "door_lock_command" and extract the signal "door_lock_activate". This way, it is not necessary for a signal to be distributed in several messages to the different nodes in order to address them separately.

Signal Deadlines

One of the critical signal properties is the timing requirement. Signals need to be delivered in a specific time frame in order to meet the requirement of low response times.

Let's look at the Mirror Control. The driver has to be able to set the position of the outside mirror on the driver's side as well as on the passenger side. Pressing the button must activate the movement of the mirror in a fair amount of time – otherwise the driver might think the mirror control is defective and releases the button and presses it again. A harder deadline is necessary when the operator is releasing the button. The mirror has to stop immediately at the position the driver desires.

This means the actuation of the button is treated differently from the release, this has to be taken into account when the signal properties are being calculated. Into the calculation of the latency times following factors have to be taken into account:

- Generation Latency

This is the time, which is needed to create the digital signal from the event or value change.

- Scheduling Latency

The time, between two instances of the same signal.

- Message Length

Time to data and overhead of a message.

- Notification Latency

Time to completely receive a message and check its consistence.

- Consumption Latency

  Time to process the message and start an action (e.g. motor starts moving)

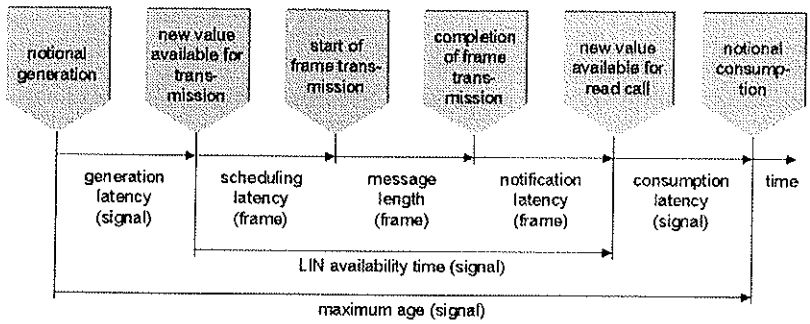

Fig. 5. Signal Latencies

Especially in complex networks (see Figure 6) with more than one hierarchy level, deadlines become a major task to deal with. Signals have to travel from one subnet via a gateway to the backbone network an
d then routed to another subnet. In this transmission the latency time of each network has to be taken into account.

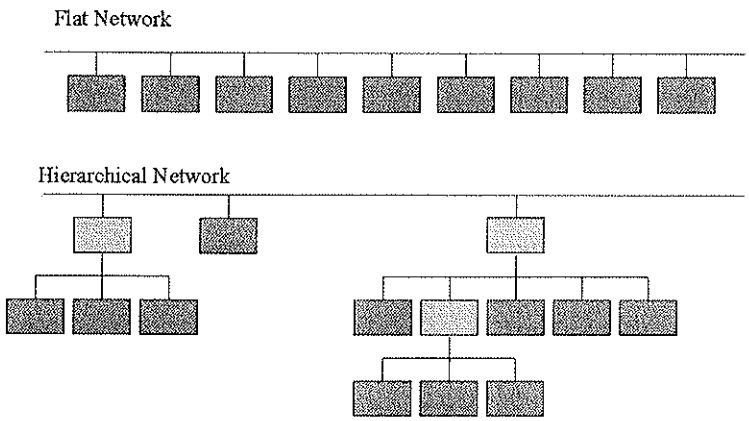

Fig. 6. Basic Network Topologies

This requires powerful tools that are capable to operate in the different network domains and can handle the some 1000 signals that travel between the different nodes of a car. A manual handling of all these messages and nodes without a proper tool support can be regarded as an impossible task.

Signal Packing

Signals are packed into messages. This is done in a way that the requirements of the signals can be met. Messages can be received by more than one node, therefore it is possible to pack signals with the same source but different destinations into one message. In our example we would e.g. pack all signals from the keyboard into one message. The message would address all other nodes, because it carries information for their applications.

The signal packing is efficiently done with tools such as the Signal Database Manager for LIN (SDML) from Volcano Communications Technologies (VCT, see Figure 7). These tools deal with all important signal properties such as deadlines, source, destination, meaning and digital representation and are capable to generate a LIN Configuration Description File, which is the basis of the LIN sub-network.

The Configuration Description File is used as input for the communication part of the nodes. It contains the information, which nodes participate in the network,

what signals they publish and subscribe how the signals are represented in the messages and what messages are transported on the bus. It also contains the schedule table, in which the order of the messages and the distance between each other are defined. The scheduling table reflects the timing requirements of the different signals in the network. The Configuration Description File does not contain information of the application, thus it serves as an abstraction/interface to the other nodes of the network.

6 Dealing with Options

Car OEMs offer their vehicles with different set of features. The door of one car is e.g. equipped with electric window lift the other one is not. The management of variants is expensive. In traditional designs variants mean different wiring harnesses - another harness for each configuration. In LIN, which communicates via a single wire, the harness doesn't change. However, the signals in the messages on the bus have to reflect the difference, but the costs are considerably lower than managing a number of harnesses. This gives the OEM more flexibility in offering more features for their cars in a cost-effective way.

7 Development Process

First Step: Capturing of the signals and their properties. In this stage the complete information flow between the nodes has to be analysed.
The necessary signals and all their properties have to be put into a database including all optional signals. This includes the source of a signal and their destinations, the maximum latency times, how the signals are represented (analogue/digital, resolution), and their dependencies. This data is ideally being processed by a tool to generate the specific configurations for all networks. Also the requirements for the different signals can be captured in a very convenient way with appropriate tools. We talk here about signals being published by a node and subscribed by other nodes.

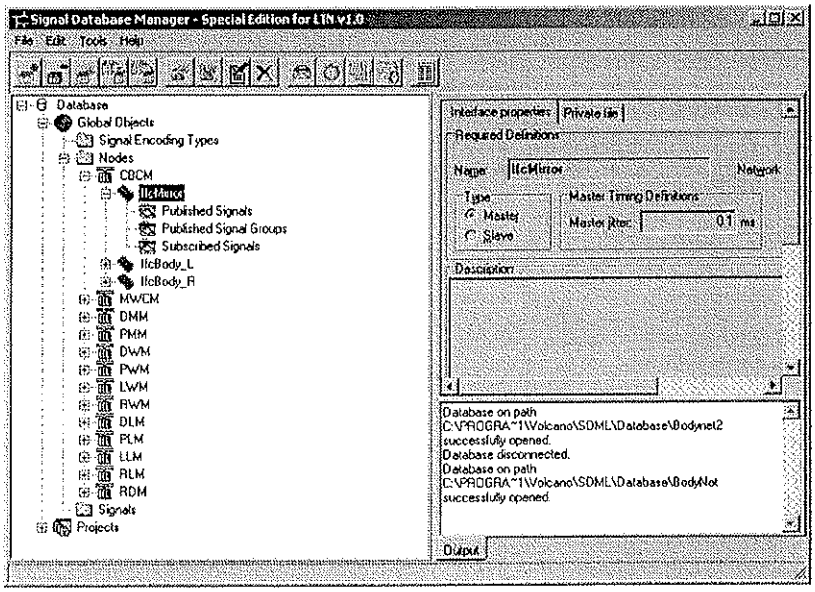

Fig. 7. Signal Definition with a Windows based Tool (VCT)

These tools are very convenient in the usage and are capable to detect timing constraints and other issues with the signal properties. Thus the designer is able to see in an early stage of the design process, if the proposed network layout can meet the requirements of the applications.

As an output the tool generates the LIN Configuration Description File, which contains all necessary network information of the particular subnet.

It is of course possible to create the Configuration Description File manually. For smaller networks this should not be too complicated. For more complex networks this can be an almost impossible task.

Node Development

The Configuration Description File serves as input for the communication part of the node software. Together with the hardware information it is been input into a Configuration tool which creates the communication module. This module is compiled and linked together with the application software.

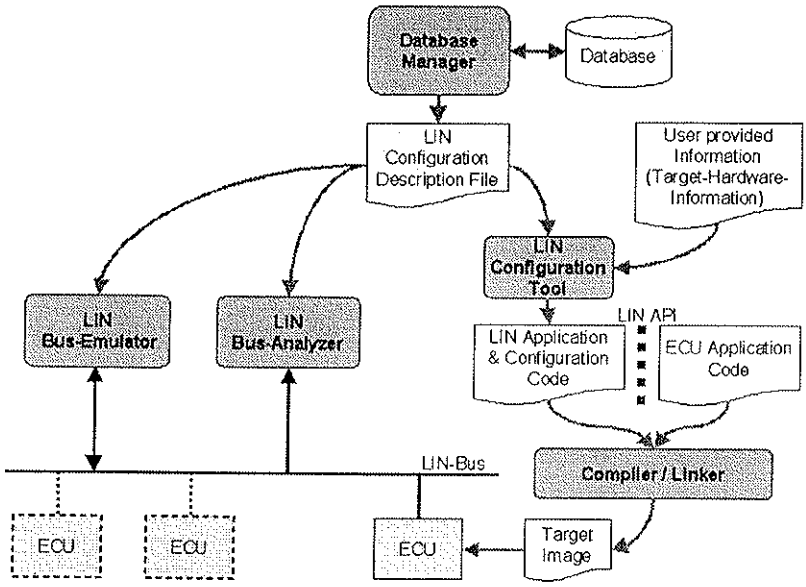

Fig. 8. LIN Development Flow

Network Emulation

The LIN Configuration Description File can also serve as information base for other development tools, such as LIN bus-emulators or LIN bus-analyzers. During the development process not all nodes will be available at the same time. In order to test single nodes the emulation of the missing nodes is required. Analysis and emulation tools from different network tool vendors (e.g. VCT and Vector) can be used to fulfil this task.

This also enables a distributed development of the different nodes in the network. Especially car OEMs, who are the owner of the network, will not create all sub-nodes. Development will take place at different subcontractors. The LIN Configuration Description File is the communication means between the partners in the project. All necessary information is passed along in the Configuration Description File.

8 Messaging Strategy

In the Scheduling Table the message strategy is reflected.

Let's look back at the mirror control problem. We have a number of signals to be transferred on the bus. Among these signals we have the activation signals to

position the mirror. In our example we pack the signals for x- and y-movement into one message frame. Signals with the same source and destination and dependencies should be in the same message.

The scheduling of the messages could look as in Figure 9: The messages occur only once in a scheduling cycle.

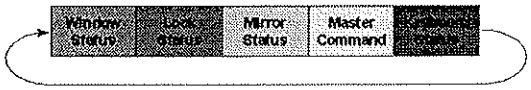

Fig. 9. Simple Scheduling

The requirement of a fast response on releasing the button is fulfilled with a dynamic scheduling table: When the key is pressed the simple scheduling table in Figure 9 is changed to the scheduling table of Figure 10. The result is a table with a much higher frequency of the keyboard status message. This way the release of the key is captured much faster and the latency time for this task is reduced.

Fig. 10. Modified Scheduling to meet stringent requirements of specific signal timings

9 Conclusion

The door example shows that LIN is capable to address the requirements of an automotive development. Timing issues can be well addressed. The development environment allows a quick evaluation of the proposed network architecture and uncovers issues at an early stage. LIN offers a high flexibility for different system configurations resulting in a cost-effective management of variants of sub-systems. The well-defined interfaces to tools and applications make it an easy task to integrate LIN into the complete car network system. However, the increasing number of nodes, network systems, and message traffic make the car multiplex network a highly complex system that cannot be managed without the help of appropriate network tools.

Acknowledgements

We thank mechatronics department at Motorola Munich for their contributions to this paper.

References

[1] LIN Consortium, "LIN Specification, Version 1.2", www.lin-subbus.org, November 2000

[2] Will Specks, Antal Rajnák, "LIN - Protocol, Development Tools, and Software Interfaces for Local Interconnect Networks in Vehicles", 9th International Conference on Electronic Systems for Vehicles, Baden-Baden, Oct. 5/6, 2000

[3] Hans-Chr. v. d. Wense, "Introduction to LIN", SAE Convention 2000, March 6-9, 2000

Additional Sources

The LIN-Specification Package and further Background information about LIN and the LIN-Consortium is available via the URL: http://www.lin-subbus.org.

Footnotes

1) MOST is a registered trademark of OASIS Silicon Systems AG.

2) FireWire is a registered trademark of Apple Computer, Inc.

3) TTP is a registered trademark of FTS Computertechnik Ges.m.b.H.

4) Flexray is a registered trademark of DaimlerChrysler AG.

5) Byteflight is a registered trademark of BMW AG.

6) LIN is a registered trademark of Motorola GmbH.

7) Bluetooth is a registered trademark of Telefonaktiebolaget LM Ericsson.

All other trademarks are registered by their respective owners.

ENGINEERING LIBRARY

Stanford University Libraries
3 6105 110 419 897

APR 25 2002

DATE DUE

NOV 21 2016