

icc 1995

2nd international CAN Conference

in London (United Kingdom)

Sponsored by

**Motorola Semiconductor
National Semiconductor
Philips Semiconductors**

Organized by

CAN in Automation (CiA)

international users and manufacturers group

Am Weichselgarten 26

D-91058 Erlangen

Phone +49-9131-69086-0

Fax +49-9131-69086-79

Email: headquarters@can-cia.de

URL: <http://www.can-cia.de>

Worldwide Status of CAN - Present and Future

Abstract

CAN network protocol originally had been invented for automotive applications. Because of its characteristics, its robustness in conjunction with its excellent performance/price ratio CAN very soon was adopted for industrial control applications widely. The high sales quantities of more than 9 mio units until 1994 and the still more promising forecast illustrate the importance of CAN as a protocol in fieldbus and sensor/actuator bus applications. The following paper discusses the historical evolution of CAN until today, giving the reasons for the big success of CAN. In a second paragraph the technical characteristics of today's CAN systems are analyzed and explained. Paragraph 3 dares a prognosis how and where CAN systems might go in the future.

1. CAN from Yesterday to Today

In the early eighties CAN = „Controller Area Network“ communication protocol had been specified by Robert Bosch GmbH for real time critical applications in cars. From 1985 on it was jointly developed with Intel Corp. until it became a first product in 1989 with the so called „FullCAN“-chip 82526 from Intel. In the mean time any of the major semiconductor supplier is already producing CAN or - at least - provides CAN samples, ref. to fig. 3.

1993/94 CAN was adopted as worldwide standard by the International Standardization Organization ISO, defining the ISO-OSI-Layers 1 and 2. As the first CAN idea was driven by the need for a new protocol for time critical communications between various controllers in a car, such as engine control, gear box control, stability control, etc., the corresponding requirements led to the so called „CAN High Speed Standard“ [1]. Another communication in cars is related to the so called „body control“ application, such as seat positioning, mirror positioning, lights on/off, switch status, etc. Influenced by the more critical cost constraints in this application field the line transmission part - that is the „lower“ part of the OSI-layer 1 - was slightly redefined, leading to the almost identical „CAN Low Speed Standard“ [2]. Any of the today's CAN chips - see fig. 3 - implements both standards. The difference comes with the line drivers only.

According to the ISO Standard the more pragmatic „CAN Specification Version 2.0“ [3] serves as the today's basis for the various CAN implementations. This specification comprises:

- Part A describing the so called „standard CAN“ message format with its 11 bit identifier, as it is defined in its predecessor, the „CAN Specification Version 1.2“
- Part B describing both the standard and the so called „extended CAN“ message formats, expanding the ISO standards towards an 29 bit identifier.

In the mean time further standardization activities are currently under their way such as the definition of a standard for CAN in agricultural application, in the US CAN has been adopted as a standard for communications in truck and busses, etc.

CAN first time was applied in Mercedes S-class cars, launched in 1992, providing a high speed network for communication between engine controller, gear box controller and dash board and a low speed network for distributed air conditioning control. BMW, Porsche, Jaguar, etc. applied CAN short time later, too. In the mean time Volkswagen, Fiat, Renault, PSA and others have decided to apply CAN, too. Today almost any of the western world car manufacturers at least

¹ Professor at Fachhochschule Wolfenbüttel, Germany, Tel.: +49 5331 939-640
Consultant to C'est Wolfenbüttel, Tel.: +49 5331 77720,

consider seriously the introduction of CAN. In the US a special task force for CAN has been established since March 1995 by the SAE = Society of Automotive engineers to investigate seriously the common needs for CAN in passenger cars.

Although in the first years of the existence of CAN chips the semiconductor manufacturers did not pay too much attentions to the industrial automation application area, many of those companies became highly interested in CAN, while looking for a low cost, high performance and robust communication protocol for their fieldbus and/or sensor/actuator-bus communication needs. Comparing the communication requirements of the machinery industries in this field, see fig. 1, with the technical features provided by CAN, see paragraph 2, easily explains why.

Characteristics Required	Priority
• Time synchronous communication:	
• required:	64%
• not required:	36%
• Sampling mode for network nodes:	
• Cyclic sampling:	73%
• Event driven sampling:	27%
• Cycle periodicity:	
• 1 ms:	26%
• approx. 10 ms:	48%
• 100 ms:	26%
• Signal response time:	
• 1 ms:	40%
• approx. 10 ms:	34%
• 100 ms:	26%
• Length of fieldbus system:	
• < 100 m:	66%
• < 1000 m:	31%
• > 1000 m:	3%

Extract of a questionnaire on fieldbus communication requirements done by VDMA, Germany in 1991 - see [4], pp. 14 for more details. 237 companies in machinery automation industries were asked for their need, representing:

- approx. 15% of the companies with < 100 employees
- approx. 39% of the companies with 100 to 500 employees
- approx. 46% of the companies with > 500 employees

Fig. 1: Communication System Needs in Machinery Automation

Although CAN turned out to be of high interest as a communication protocol in machinery automation industries some additional harmonization efforts and extensions were needed to raise CAN based systems more towards the „Open Systems“-idea, easing intercommunication in non closed, non proprietary multi vendor systems. This was the situation when CiA = „CAN in Automation“ users club [4] started in 1992. Today CiA comprises approx. 200 members: semiconductor suppliers, machinery automization suppliers and manufacturers, automotive suppliers and manufacturers, tools suppliers, etc. In the beginning the members mostly came from Germany, while in the mean time almost any European country and the US are actively cooperating.

The great success of the CiA is due to the fact that the related working groups very quickly worked out pragmatic solutions for the open questions with regard to the existing CAN standard, such as [6]:

- Narrow down the OSI-layer 1 signal transmission specification to:
 - cable recommendation
 - connector recommendation
 - transceiver recommendation
- Specification of a solution for the lacking OSI-layer 7 interface to the application itself. As a first solution very soon CAL = „CAN Application Layer“ was specified, supported very much by Philips.

Based on the resulting „Common Agreements“ in the CiA community the already existing CAN Standard was extended by „Quasi-CiA-Standard-Complements“, expanding CAN protocol towards an „Open System“. The resulting interoperability was the break through for the adoption of CAN in the industrial automation industries. Other than in the automotive industries, where each of the „big“ auto manufacturers would² be in the position to specify the low end OSI-layer 1 and the OSI-layer 7 by its own because of the high quantity of communication nodes, most of the industrial automation companies are small and medium enterprises and therefore they are very much dependent on an established „standard“ allowing:

- Multi vendor systems, giving their customers more safety and freedom to put a system together, consisting of modules from different suppliers, such as:
 - PLC central control from supplier x_1 or x_2 or ...
 - Intelligent motor control from supplier y_1 or y_2 or ...
 - Intelligent process monitor from supplier z_1 or z_2 or ...
 - Installation and test tool from supplier t_1 or t_2 or ...
 -
- Reduction of costs due to no/simpler adaptation of generic proprietary hardware/software investments towards a customized system, etc.

Today CiA is successfully continuing to work on the harmonization of open question related to CAN in industrial automation, thereby broadening the acceptance of CAN and thereby expanding the market for the CAN community by :

- Providing a central authority for press requests
- Organizing common CAN presentations at fairs, conferences, workshops, etc.
- Building bridges to other CAN activities such as automotive applications
- Providing a platform for further technical discussions and decisions to achieve specification harmonization of ideas such as:
 - Further OSI-layer 7-protocols such as: SDS, DeviceNet,
 - Profiles for various applications
 - Testing, validation, certification of interfaces

In the mean time CAN has become already a great success, looking at the already achieved sales, see fig. 2. Since its begin in 1989 until end 1994 already more than 9 million CAN interfaces have been sold. Referring to the commitments, which have been given to the semiconductor manufacturers until today, in the years 1995 and 1996 additional more than 20 million CAN interfaces will be sold, adding up to a total of more than 30 million. Based on that until the end of this century more than a total of 140 million will have been sold. These figures have been derived from all CAN semiconductor manufacturers inquiries. They are put together in rather a conservative way, as for instance some recent decision for CAN application by PSA, Renault are not included to their full extent and the figures do not reflect the industrial potential of the CAN market evolution to be expected in the industrial automization market especially in the US.

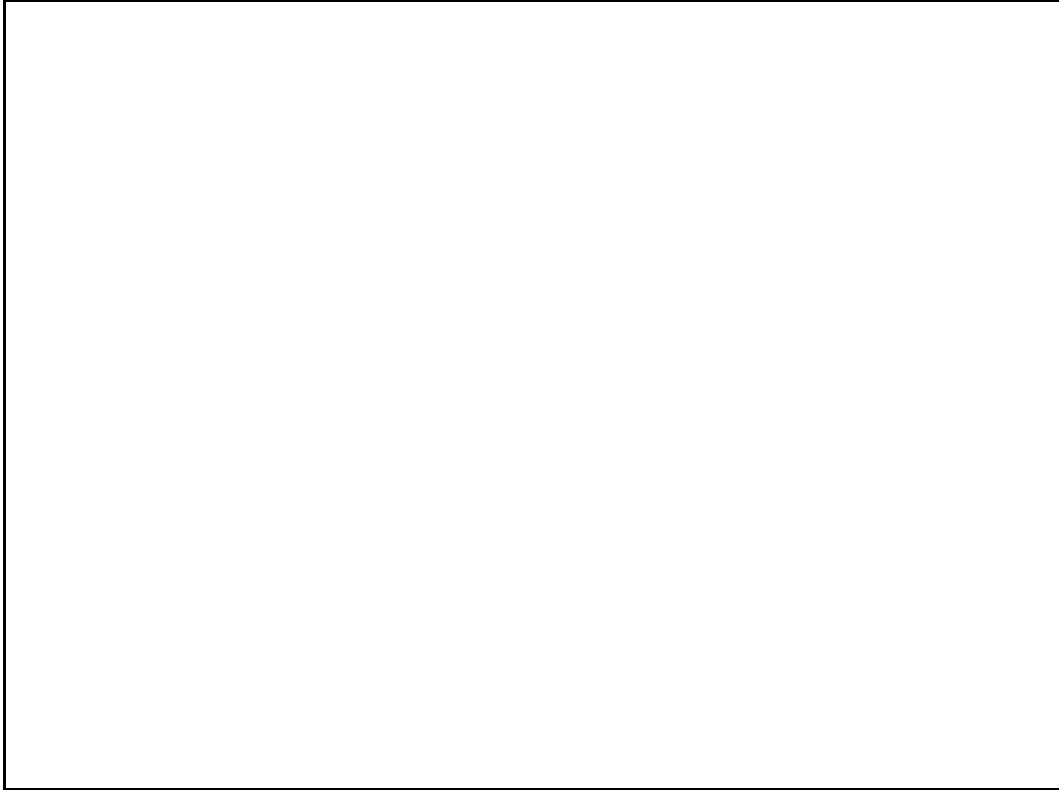
Looking at the conservative CAN sales figures in fig. 2 it can be easily stated that the break through for CAN has already been passed since 1993/94. CAN has become „very real“. The CAN sales gradient is steadily raising:

- At 1995 approx. 10 mio units per year are sold
- At 1996 approx. 20 mio units per year will be sold
- At 1999 approx. 40 mio units per year are expected to be sold.
- Beyond 1999 still a raising increase rate of the CAN sales can be expected because of the rapidly increasing CAN application in the high volume demanding automotive business, especially taking into account their long lead time of more than 3 years from design to production of a new model.
- Further increase even before 1999 can be expected from the developments mentioned above.

The majority of the CAN sales status and forecast still comes from automotive application. But this is one of the strong reasons to go with CAN; this is like a kind of „life insurence“ while

² Since 1993/94/95 more and more cars manufacuturers gave up their proprietary communication protocol solution while switching to CAN, expecting by this decision further increase of the CAN production volume and thus reducing the costs per communication node.

deciding for CAN. Because of the big quantities applied in automobiles the industrial CAN applications will benefit automatically by:



- Multiple sources, see fig. 3
- Low costs; in automotive industries the price target is 1 US\$ per CAN node at high quantities
- High performance for low price
- High proliferation rate, driven by automotive industries

Recognizing these reasons and/or appreciating the already today's excellent cost/performance-ratio CAN has been adopted in many of the industrial automation control fields as the communication protocol for fieldbus and/or Sensor/Actuator-bus applications. Almost all of the big industrial control companies and very many of the medium and small enterprises offer CAN interfaces for their systems in a variety of application areas:

- Vehicles with CAN Communication such as:
Passenger Cars, Trucks, Air Planes, Trains, Boats, Agriculture Machines, Excavators, Pavement Machines, etc.
- Industrial Control with CAN Communication such as:
Programmable Logic Control (PLC), Robot Control, Intelligent Motor Control, Intelligent IO - Sensors/Actuators, Hydraulics, Intelligent Sensors/Counters for various applications such as Water Consumption, Electric Power consumption, etc., Textile: Spinning, Weaving, etc., Medical, Laboratory Automization, Buildings, Elevators, Vending/Teller Machines, Toys, Mechanical Tools, etc.

2. CAN Technology Today

2.1 Generic CAN Characteristics

The generic CAN features due to their OSI-layers 1 and 2 specifications, as derived from the standard [1], [2] and the actual specification [3] provide a very powerful basis for communication, summarized as follows:

- The CAN multi master hierarchy feature has been chosen for safety and availability reasons: If a node fails, the whole system does not collapse, the system performance is gracefully degraded.
- The very powerful error handling technique with a Hamming Distance of 6 insures the

from the network, in case the node detects itself to be - very likely - malfunctioning. Thus total system break down is prevented, if only one (or more) node(s) are defect. In case of network problems system recovery time is deterministic.

- In correspondence to the multi master feature CAN performs priority based CSMA/CR = Carrier Sense, Multiple Access/Collision Resolution media access technique: Any node, which tries to access the communication media waits until the bus is idle and then starts communication by transmitting a start bit first (for synchronization of all partners), followed by the 11³ so called Identifier bits, representing the name and the priority of a message, and then followed by control bits (6), data (0 to 8 byte), CRC (16 bits), Acknowledge (2 bits) and - finally - the end of frame (7 bits) and the inter frame space (3 bits). In case more than 1 node simultaneously tries to access the bus, the collision is resolved by a bit wise arbitration process, carried out over the identifier field. This guarantees the highest priority message to be transmitted first without any additional delay and then the next lower priority messages will follow in the order of their priority. After a message frame has terminated its transmission, a new round of prioritized bus access starts.
- Communication generically is event driven. For instance on the event that a node has produced new information this node can initiate the transmission of that information itself. If a cyclic transmission is required, this self initiative can be substituted by a cyclic driven one, initiated by an internal clock or by another (master) node or ... These philosophies require corresponding software either done in the application program or, which is the better solution, provided by an „OSI-layer-7“ application interface, see paragraph 2.3⁴.
- A CAN network provides high flexibility in system lay out:
 - Number of Nodes/System:
 - 32 with standard line drivers
 - 64 ... 128 ... with application specific line drivers; see e.g. SDS drivers from Honeywell, [11]
 - Number of Messages/System:
 - 2¹¹ with Standard CAN 2.0A
 - 2²⁹ with Extended CAN 2.0B
 - Number of Data/Message: 0 to 8 Byte
 - Gross Maximum Length/Message:
 - 117 Bits with Standard CAN 2.0A
 - 136 bits with Extended CAN 2.0B
 - Bitrate-Bus-Length Relation⁵:
 Bus Length * Bit Rate < 40 m * 1 Mbit/s (Assuming < 25 ns propagation delay/transceiver)
 that is approx. e.g.:

40 m at	1 MBit/s (max. Bitrate)
400 m at	100 KBit/s
1000 m at	40 Kbit/s
- There is a high variety of CAN chip vendors, such as Intel, Motorola, Philips, Siemens, NSC, NEC, etc.; fig. 3. This prevents the customers from the second source problem and insures good prices and high proliferation rates, see fig. 6. Today CAN chips are available with:
 - Various message buffer size on board and methodology (so called BasicCAN, FullCAN)
 - Various additional intelligence such as time stamps for received messages, etc.
 - Various integration degree:
 - Stand alone, requiring additional micro controller and line driver
 - Integrated solution with micro controller on board, requiring additional line driver only
 - Super integrated solution with everything on board, incl. power supply (available soon)
 - Low scale integrated solution with parameterized logic, interfacing directly registered IO, requiring additional line driver only
 - Various (VHDL) ASIC CAN cells for customer specific integrated solutions

Character/Supplier	Parts Name	Protocol Type	Package Size	Availability	Degree of Integration	On Board _ C/ Interface to _ C	Mailboxes Type, No.	Miscellaneous
--------------------	------------	---------------	--------------	--------------	-----------------------	--------------------------------	---------------------	---------------

³ 11 bit for Standard CAN 2.0A, 29 bit for Extended CAN 2.0B

⁴ SLIO-CAN provides a limited automatic transmission capability on the occasion of special events - such as an input exceeds a given threshold - without requiring any micro controller action.

INTEL	82527	CAN 2.0B	44 PLCC 44 QFP	Production Prod Q4/95	Standalone	serial, 8 / 16 bit multiplex/non-m.	FullCAN, 14 BasicCAN, 1	2 Acceptance Mask Filter: Two 8-bit I/O ports
INTEL	87C196CA	CAN 2.0B	68 PLCC	Production	On board μ C	16-bit _ C 32K ROM 1.25K RAM	FullCAN, 14 BasicCAN, 1	6 high speed I/O channels, 6 A/D inputs, SSIO/SIO
INTEL	87C196CB	CAN 2.0B	84 PLCC 100 QFP	Samples, Prod Q3/95 Spl. Q3/95	On board μ C	16-bit _ C 56K ROM 2K RAM	FullCAN, 14 BasicCAN, 1	10 high speed I/O chan., 8 A/D inputs, SSIO/SIO
INTER- METALL	Custom Dashboard Controller	CAN 2.0B	80 QFP 100 QFP	Samples	On board μ C	65C02 48K ROM 1.5K RAM	FullCAN, 15 BasicCAN, 14	Time stamp at each mailb More mailboxes at lower speed. BasicCAN with variable No. of Rx-Buffers
MOTOROL A	68HC05X4	CAN 1.2	28 SOIC	Production	On board μ C	8-bit _ C 4K ROM 176 Bytes RAM	BasicCAN	16 I/O ports 16 bit timer, 1I/C, 1 O/C
MOTOROL A	68HC705X4	CAN 1.2	28 SOIC	Production	On board μ C	8-bit _ C 4K EPROM 176 Bytes RAM	BasicCAN	16 I/O ports 16 bit timer 1I/C, 1 O/C
MOTOROL A	68HC05X16	CAN 1.2	64 QFP	Production	On board μ C	8-bit _ C 16K ROM 256 Bytes RAM	BasicCAN	256 Bytes EEPROM, 16 bit counter, 2 Pulse Length Modulation chan., 2 input capture channels, 2 output compare chan., 8 A/D 8-bit input channels,
MOTOROL A	68HC05X32	CAN 1.2	64 QFP	Production	On board μ C	8-bit _ C 32K ROM 528 Bytes RAM	BasicCAN	Auto transmiss. on signal transition. All CAN IDs available. IO-pins program. as analog/digital
NSC	MM 57C360	CAN 2.0 A 2.0 B pass.	28 SO	Production	SLIO	No _ C No interface	14 Bit I/O-Register	IDs available. IO-pins program. as analog/digital
NSC	MM 57C362	CAN 2.0 A 2.0 B pass.	20 SO	Production	SLIO	No _ C No interface	6 Bit I/O Register	
NSC	COP884BC	CAN 2.0 A and 2.0 B passive	28 SO	Samples	On board μ C	8-bit _ C 2K ROM 64 Bytes RAM	BasicCAN, 1	Reduced BasicCAN: 7-bit CAN-Mask, Messgs longer than 2 Byte Data require additional software
	COP87M84BC = OTP version of COP 884 BC							
PHILIPS	82C200	CAN 2.0 A	28 DIP 28 SO	Production Production	Standalone	8 bit multiplex. Intel / Motorola	Basic CAN, 1	
PHILIPS	P8xC592	CAN 2.0 A	68 PLCC	Production	On board μ C	8-bit _ C 16K (E)PROM 2X256 RAM	BasicCAN	5 X 8-bit I/O ports, DMA for CAN messages
PHILIPS	P8xC598	CAN 2.0 A	80 QFP	Production	On board μ C	8-bit _ C 32K (E)PROM 2X256 RAM	Basic CAN	5 X 8-bit I/O ports, DMA for CAN messages
PHILIPS	P82C150	CAN 2.0 A and 2.0 B passive	28 SO	Production	SLIO	No _ C No interface	2X8 Bit IO-Registers	I/O-pins configurable as ≤ 16 digital I/O ≤ 6 analog, 10-bit resol.
SIEMENS	81C91	CAN 2.0 B passive	28 PLCC	Production	Standalone	8 bit multiplexed or serial interface	FullCAN, 15 BasicCAN, 1	8 Time stamps assigned to mailboxes. Message ID reprogrammable any time
SIEMENS	81C90	CAN 2.0 B passive	44 PLCC	Production	Standalone	8 bit multiplexed or serial interface	FullCAN, 15 BasicCAN, 1	8 Time stamps assigned to mailboxes. 2X 8-bit port extensions
SIEMENS	C167C	CAN 2.0 B	144 PMQFP	Production	On board μ C	16-bit _ C 8K ROM 2-4K RAM	FullCAN, 15 BasicCAN, 1	Message IDs reprogrammable at any time.
SIEMENS	C167CR- 16F	CAN 2.0 B	144 PMQFP	Samples in Q2/95	On board μ C	16-bit _ C 8K ROM 2-4K RAM	FullCAN, 15 BasicCAN, 1	See C167 C
SGS	ST10F167	CAN 2.0 B	144 PLCC	Samples in Q2/95	On board μ C	16-bit _ C	FullCAN BasicCAN	Compatible to Siemens C167C
TEMIC	29C464	CAN 2.0 B	52 QFP 52 PLCC	Samples in Q3/96	On board μ C	8-bit μC, 16/32K ROM 512 RAM	FullCAN, 14	1 mask/mailbox, 2x16 bits timers, SPI, 8 bits A/D, 5x16 bits IC/OC/PWM, UART, keyboard port
TEMIC	29C461	CAN 2.0 B	28 SO 28 PLCC	Samples in Q3/96	Stand alone	SPI or 8/16 bit multiplex./non m.	FullCAN, 14	1 mask/mailbox
TEXAS Instruments	TI CAN- Module	CAN 2.0 A and 2.0 B	NA	Q4/95	IC - CELL	8/16-bit non- multiplexed	FullCAN, 16	Transmission time-out Low power mode. Bus error diagnostic
TEXAS Instruments	TMS 370E08D55	CAN 2.0 A and 2.0 B	64 QFP 68 PLCC	Samples 1Q '96	On board μ C	8-bit μC 16K EPROM 1K RAM	FullCAN, 16	Watchdog RTI, 2 Timers PWM, 24 I/O Bits, I ² C, SCI, A/D 8 bit

Fig. 3: CAN-Module Characteristics and Suppliers
2.2 CAN OSI-Layer 1 and 2 Implementations

Today a huge variety of CAN implementations for the OSI-layers 1 and 2 is available on the market. Almost any of the world's semiconductor manufacturers has already a CAN license signed with Bosch and/or provides CAN chips at least as samples. The various implementations can be classified by the parameters CAN protocol version, CAN internal mailbox structure, CAN integration degree, CAN interface to processor and CAN interface to communication media - more details to that classification are given and considered in paragraph 3, ref. to fig. 6.

Fig. 3 gives an overview on the today's existing CAN controller chips, most of them are in production since several years, although in recent times many of additional CAN components were launched, indicating the huge demand existing or expected from the market today or in the near future. The broad choice of the different implementations is - again - driven by various, specific high volume applications, focusing at an application specific high performance/price effectiveness, each. Therefore these solutions provide the advantage, too, for the industrial control customers to do their own choice according to optimize the cost/performance ratio for increased product competitiveness.

Fig. 4 lists the today's existing CAN-transceiver solutions. The solutions are compliant to the standard and the CAN 2.0 specification although they vary in many additional features as mentioned in fig. 6. A lot of efforts were made to compete with the harsh EMC requirements for cars. Therefore the various transceivers differ in their slope smoothing capabilities while trying to reduce the amount of additionally required external components and thus trying to find an acceptable overall cost minimum. Smoothing the edges reduces the higher frequency components contained transmitted CAN signal while travelling over the lines and thus reduces radiated noise. On the other side smoothed signal edges reduce the line termination problem according to the reflection problem, but it reduces on the other hand the max. achievable line length for a given bit rate. These are some of the problems coming along with the definition of line transceivers and thus demonstrating the overall problem's complexity. On the other hand it is rather easy to realize a solution for a specific application as long as the responsible designer fully obeys the constraints, given with the applied components.

Today there is no „integrated“ solution available, having transceiver and CAN controller on the same chip. This is due to the technical problems while integrating on the same chip logic cells and power cells, while having in mind that for automotive applications the high immunity requirements against external disturbances such as overvoltage spikes in the range of up to more than a hundred Volts create problems. Never the less in the future probably there will be integrated versions available, coming with the technology progress. Already today Motorola provides an very high scale integrated solution containing a J1850 communication controller, an application processor, the line transceiver, some IO-registers and a voltage regulator on the same chip.

- 80C250, 80C251; Philips
- SN 75 LBC 031; Texas Instruments (announced)
- B10011S; TEMIC (announced)
- MTC 3054; Alcatel Mietec (announced)
- CF150B; Bosch Microelectronics
- Customized dedicated transceivers, such as SDS driver from Honeywell with very low quiescent current, capable of driving up to 64 or 128 nodes, etc.

Fig. 4: CAN Transceiver

2.3 CAN OSI-Layer 7 Implementations

Obviously any application with CAN communication can be implemented on the basis of the above mentioned components. But this requires many detailed operation of the software designer as far as the CAN communication itself is concerned. Before transmitting a newly read sensor value over the CAN network the variable first must be written into the appropriate CAN

the transmission was successfully terminated, etc. As these actions - „Transmission“ and „Reception“ of messages, accordingly accompanied by checks and the related exception handling - typically are quasi standard actions, that need to be performed in a similar manner very often, the application software designers, after getting bored with the low end details, very soon start writing their own „optimal“ subroutines or so called „software drivers“ for those purposes.

This indicates an important issue: „Transmit/Receive/Exception-Handler/etc.“-Software-Drivers are needed in order to hide the details from the application software programmer! Drivers significantly reduce the costs and time for programming, while increasing the quality of the product.

But very often this idea is not thought through until its final extent: Writing its own proprietary software driver very often is under estimated according to the related costs for its production and its maintenance. Very often engineers think that they might be able in a short time to write their own „optimal“ versions with low consumption of RAM, ROM, processor power and throughput time. But very soon it turns out that additional cases and exceptions need to be considered and thus the amount of required man-power and time increases more and more. On the other hand the so called „interoperability“ or the „System’s Openness“ apparently can not be achieved, if such proprietary „optimal“ solutions are pursued and applied.

And those are the reasons which justify the harmonization efforts to specify a common „OSI-Layer 7“ or „Application Layer“ interface. One of the major steps when starting the CiA, was the definition of CAL = CAN Application Layer. In the mean time some further CAN application layer systems have been developed, some of them have been in existence before, see fig. 5

- CAL = CAN Application Layer
From CiA Users Group, Multiple Sources: For Distributed Control
- SDS = Smart Distributed System
From Honeywell: For Distributed Sensor/Actuator Link
- DeviceNet
From Allen Bradley: For Distributed Sensor/Actuator/PLC Link
- CAN Kingdom
From Kvasar: For Distributed Control
- J 1939
From SAE = Society of Automotive Engineers (USA): For Trucks
- OSEK = Open Systems and Interfaces for Distributed Electronics in Cars
From European Car Industries Users Group: For Cars

Fig. 5: CAN Application Layers

Looking at the big variety of higher layers listed, the question may arise, WHY is there a need for so many different application layers, not being compatible among each others? On one side the various approaches are justifying themselves by requiring different amount of computer resources such as ROM size, RAM size, processor power and through put time. The difference mainly is due to the different optimization assessments, focusing on the amount of services offered, the flexibility according to variables handling, network management facilities, flexibility in terms of broader or smaller variety of CAN chips supported, processor types and operating systems supported. On the other side very often historical reasons, market competition with much money involved and thus highly political and other reasons have big influence.

Looking at the technical differences for instance CAL is providing the biggest support as the above mentioned constraints are concerned and therefore it is applicable for complex distributed systems solutions. But - as a consequence - CAL is the most resources consuming implementation. SDS and DeviceNet are very dedicated higher layer protocols, being highly efficient in terms of computer resource consumption, when applied for more simple distributed

pure communication services, OSEK incorporates the network management issues and an integrated vision of the interface towards a real time multi tasking operating system. OSEK today still is in the definition phase. For more details refer to [4], [5], [6], [7], [10], [11], [12], [13].

As a simple summary, each of the different higher layer protocols is not better nor worse than another one from a generic, technical point of view. Before judging, the needs from the final application point of view must be analyzed. Comparing these needs with the specific characteristics of each higher layer protocol then results in a ranking.

This selection process is easier said than done in reality due to the many parameters involved. On the other hand - see paragraph 3 - hopefully there will not be too many higher layer protocols thus guaranteeing high degree of interoperability, multiple sources, good price/performance ratio and good maintenance capabilities.

2.4 CAN Development and Test Techniques and Tools

„What does a screw help of there is no methodology how to apply the screw and if there is no screw driver“!

This very often explains the problem coming along with new technologies and new components: A priori there is not enough information available supporting the application of the new component, nor are there efficient tools available, supporting the different life phases of the component [8], [14]. How can a very high complex distributed system - as it typically is along with a CAN network - be tested, if there is only a volt-meter available!? Such a situation would afford much supplementary and needless man-power and time. And this corresponds to more costs and lower customer satisfaction. Very often this situation is not transparent from the first moment on to the management, as engineers typically like to solve sophisticated questions by themselves and therefore they regard this problem as their special personal challenge. And all this explains why they typically report so (too) lately to the management.

According to CAN the question of sufficiently powerful and graded development and test philosophies and tools is solved! Subsequently the basic procedure while doing a new application development and test is described below, when applying CAL [6], [10] as the application layer⁶:

The first step is the definition of all CAL variables. Therefor a so called System Builder Tool [9] provides a window's like entry table from, asking any of the variables' attributes from the development engineer and performing consistency checks. At this stage the assignment of the „real“ CAN identifiers to the „symbolic“ CAN variables is performed. All this information is stored in the central system data base.

When writing the actual application code no additional specification of the communication variables must be done. These variables can be immediately referenced. When starting the compiler, the CAL driver package and the variables definition file from the system builder only must be added per include - and the work is done. This technique additionally offloads the application code programmer, while sharing the work with the system definition engineer. The resulting object code is down loaded to the target controllers and then the test phase starts.

Testing in conjunction with the System Builder is easy. The System Builder provides all the variables' definition files to the various network test tools, such as Analyzers and Emulators [9]. The test operator no more needs to refer to the physical details of the CAN variables. He can perform his task on the symbolic level. On the same „higher level“ the „life“ status of the CAN-network-nodes can be monitored and/or stimulated. Based on this idea certification and validation of new network nodes can easily can be done. In case of „lower layer“ problems the functionality of Emulators and Analyzers helps to stimulate and monitor the operation on variables down to the message and or bit level, while still accessing these entities on the symbolic level. Additional network modules can be simulated in real time, errors can be injected and the system's behavior on these additional normal/abnormal loads can be monitored and compared to the proper required system reaction. In case of problems, which

may be located at the very low end wiring side, other tools such as logic analyzers, data generators, oscilloscopes etc. can be synchronized with the higher tools operations.

As a summary: There is a well established philosophy for CAN systems development and the related tools. A well graded family of tools is existing, covering the range from starter kits over evaluation kits, PC-interfaces towards high end tools such as Emulators and Analyzers and System Builders. In addition a broad spectrum of hardware and software for process control modules is available on the market. All these items are supplied by a very huge variety of small, medium and large enterprises, thus guaranteeing continuous support of the final CAN system customers.

3. „Quo Vadis CANe“?!

Where is CAN going? - Having the today's and future CAN sales in mind - see fig. 2 - it is no question: CAN has become reality. It has already been sold in such high quantities which had not been achieved at least by some of its fieldbus-competitors, yet. The number of CAN-chip semiconductor manufacturer is overwhelmingly high. Multiple tools suppliers provide almost any support that might be required. In general the price/performance ratio of CAN and the related products is excellent. So - what needs to be done with CAN? Is there any proliferation needed and - if so - where might it go?!

The following considerations about possible future CAN evolution pragmatically are based upon and restricted to 2 empirical approaches: One is to analyze the evolution of the system/component, which is under consideration, beginning from the past until today. Find out about the past trend of the system's orthogonal parameters and do an extrapolation. This methodology of course does not allow non-continuous predictions. Another approach tries to identify another already existing system, which can serve as an advanced (future) model for the system under consideration. Mapping the features and characteristics of this model to the system under consideration may result in the prediction of non-continuous evolution directions.

According to the first assessment methodology the evolution of CAN from its first implementation - the Intel 80526 chip - until today will be analyzed and the consequences will be drawn. For the second methodology the simple architecture of a single computer system will be assessed to derive a so called „Virtually Commonly Memory“ model with its so called „Shared Variable“ to understand and simplify the complex phenomena of a networked distributed system and to derive ideas for future (CAN-) networked systems [4], [8], [14]. On the other side the following consideration are based on the assumption that semiconductor technology will evolve rapidly, as it has been doing in the past, providing some 10 times higher functionality every 3 years while keeping the price fixed. Last but not least the considerations will be led by the generic application constraints of real time, sampled control loops, which their intrinsic sensitivity for time delays in the components/network.

In the beginning the first CAN implementation was the Intel 82526 chip, providing the following characteristics: Standard-CAN with 11 bit identifier, FullCAN mailboxes, no masks, number of mailboxes varies between 5 and 14 depending on the data length of each message, flexible definition of tx/rx-mailboxes, redefinition off-line only, no time stamps, stand alone, 2x8 bit digital IO-ports, parallel multiplexed Address/Data-Bus interface to processor, no power line transceiver on board, etc. Taking this as a basis and comparing the features of the today's CAN implementations as shown in fig. 3, the „abstract parameters“ shown in fig. 6 globally characterize any of the actual CAN chips. Drawing the bottom line and reminding the changes over time, the following of changes are apparent up to today, characterizing implicitly/explicitly the future trends of CAN chips evolution:

- CAN protocol version:
Extension from Standard CAN Identifier with 11 bits towards 29 bit Identifier Extended CAN. Both systems will live; compatibility of mixed systems is partly achieved. The 29 bit identifier extension was introduced for compatibility purposes with the US SAE diagnostic standard J1978 and successors for cars' applications, specifying a 3 byte header. Standard CAN is sufficient for most of the industrial applications, reducing costs.
- Mailboxes:

masks for each of the multiple receiver mailboxes will come. For real time critical⁷ applications the time stamps will become more and more important.

- CAN integration degree:

In the beginning stand alone versions were very advantageous, because they offered greater freedom for the customer to select the processor of his choice. The trend very clearly will go towards integrated versions, as already today almost any of the major micro controller architectures is already available on the market providing CAN on the same chip. In the future CAN will become something like the today's UART, being available on almost any of the today's micro controllers already. The integrated version in addition offers easier and faster CAN register access from the micro controller.

Recently more and more dedicated, real time application oriented IO was integrated. This trend will continue, as the availability of correspondingly intelligent IOs - such as capture and compare registers, PWM, counters, etc. - offloads significantly the program overhead and at the same time enhances the real time capabilities. This is a more generic trend and not so much related to CAN networking.

SLIO = Slave IO-CAN CAN be regarded as a slimmed down integrated micro controller/CAN solution. SLIO-CAN provides very few low scale mailboxes - 6 to approx. 16 bits -, being almost directly coupled to the IO-pins of the chip. Minor changes of the IO/CAN-communication functionality CAN be achieved through limited parameterization, such as automatic transmission of an input value, if the input value has exceeded a certain threshold limit. No flexible programming facility. But low silicon size.

The SLIO implementation had been introduced as a stripped down CAN/processor solution, targeting for very low cost. Probably the performance/cost ratio of these kind of devices may not be competitive compared to the lower end, programmable integrated micro controller solutions, offering greater flexibility at a comparable price. There is the vicious circle phenomena: As micro controller based solutions have been available since long time and they are already produced in higher quantities, resulting in lower prices, attracting more customers, pushing up the production quantities and

Transceivers as well as supply voltage regulators will be integrated in the future. With technology progress there will be a better basis for integrating those different kinds of circuitry on the same die.

- Interface to processor:

Typically parallel, even DMA driven access techniques will be predominant, corresponding to the fact that the integrated solution, providing CAN interface and processor on the same chip.

- CAN interface to communication media

Many efforts will concentrate in this field in the future. Only since very recent times there are solutions on the market - Philips, Bosch, see fig. 4 - which go into the right direction according to pricing and performance. Low cost solutions with good EMC behavior and high immunity against over voltage are difficult to design. First attempts of transceivers with integrated line diagnosis are announced. A lot of progress will be expected in this field, probably very much driven by the automotive requirements and their urgent needs.

All the above tendencies were derived directly from the evolution of the parameters, describing the today's CAN implementations. Very promising solutions according to the related OSI-layer 1 and 2 CAN chip hardware solution can be expected. Looking at the scenario more from a top down view point, from the application side, the complexity seems to be over whelming: Many different CAN interfaces make it difficult or almost impossible for the users to switch from one CAN implementation to another one. The various niceties offered by each solution again may be confusing, apart from the positive aspects, discussed above. One solution from this conflict is

⁷ In general any application is a real time application. It becomes „critical“, if the required system reaction time - corresponding to the Shannon Law [15] sample rate - is in the same order of magnitude as the through put delay of the whole system, consisting of the network

the introduction of the OSI-layer 7 interface, hiding all the details coming from the various implementations.

- CAN protocol version
 - Version 2.0A („Standard CAN“, 11 bit Identifier)
 - Version 2.0B („Extended CAN“, 29 bit Identifier)
 - active
 - passive
- CAN internal mailbox structure
 - Number of mailboxes
 - Type of mailboxes
 - With ID-mask („BasicCAN“)
 - Length of mask
 - Influence of mask
 - On a single receive mailbox, each
 - On multiple/all receive buffers
 - Depth of FIFO buffers behind receive mailbox
 - Without ID-mask („FullCAN“)
 - Definition of mailboxes for reception/transmission
 - Fixed/flexible redefinition
 - on-line/off-line redefinition
 - Safeguarding of message transmission
 - Time stamps available
- CAN integration degree
 - Stand alone, processor off-board required
 - Processor on chip integrated
 - IO-registers integrated:
 - with direct CAN access („SLIO“)
 - with no direct CAN access:
 - Digital/Analog
 - PWM, etc.
 - Timers/Counters
 - Capture and Compare
 - Transceiver on chip integrated
 - Supply voltage regulator on chip integrated
- CAN interface to processor
 - Parallel Address/Data Bus
 - Multiplexed Address/Data Bus
 - SPI Address/Data Bus
 - Single word/DMA access
- CAN interface to communication media
 - power consumption
 - sleep mode
 - internal pulse shapening
 - discharge immunity
 - load dump immunity
 - line diagnosis
 - line fault tolerance

Fig. 6: CAN Structure Parameters

Unfortunately there are already many of the CAN application interfaces existing, see fig. 5. Further more there is a need for better and easier system design capabilities in real time critical applications. In addition a more uniform and more simplified approach for system design, even independent from the existing CAN OSI-layer-7-implementations might be desired. This idea might lead to a more neutral - may be unique - interface to the actual application, going beyond OSI-layer-7. A possible solution for that may be derived from the „virtual commonly shared memory“-model in conjunction with the „shared Variables“-concept, which represents an extension of the today’s known variables, which are applied in standard computer languages [4], [7], [14]. The basic ideas will be explained below and may lead in the future to new and better efficiency solutions.

The Shared Variable idea generically is already known and already applied to a certain extent in some communication protocols such as LON, P-Net, etc. The concept lives from the idea that the whole distributed system can be understood as an virtual single controller system, providing a virtual commonly shared memory serving as the exchange buffer device for the interchange of any variables, the virtually shared variables. The programmer e.g. accesses and operates on IO-variables, making no difference whether these variables are local or remote ones and making no difference whether the remote IO-variables are communicated through the network or by direct dedicated lines.

The difference between local and remote shared variables of course must be made known to the system some where and some how. In accordance to the already applied programming strategies this will be done by extending the „standard“ declaration part of the „standard“ variables, defining in a supplement, whether this is a local or remote variable and if remote, where it is located, etc. Derived from this definition, automatically - without the explicit interaction of the programmer - a call to a correspondingly specific service of the underlying CAN-OSI-layer-7-implementation is done, taking care of all the details related to the underlying CAN-chip-implementation.

One additional extension to the already extended variables is required, taking into account the constraints coming from the real time requirements of the intended application system: Each variable definition therefor is enhanced by a set of attributes reflecting these timing requirements, such as „Access Time Required“. This attribute, among others will be used by an Enhanced System Builder tool to find out about how the communication of the related variable must be done: as local IO, remote-dedicated line IO or networked IO. In case a networked IO is requested or selected, an ID assignment is done and a mode, defining how, when, how often and by whom the corresponding message transmission will be stimulated stimulated, automatically is generated, taking into account on the underlying OSI-7-layer implementation or the direct CAN chip implementation.

This idea needs further research and development. It is important to develop it further under the constraint of the strict avoidance to invent a variation of the today’s existing and currently applied computer languages. The supplementary attributes of the variables must be removable in a front end compilation process, thus allowing the application of standard compilers and thus not being stuck to a specialized language and the related compiler variation problems.

Summarizing the ideas, it was the goal to point out future ways which probably might occur to the evolution of CAN related systems in the future. There is a need for significantly more simplified application interfaces for distributed systems, making life easier for the application programmers and providing a quasi „hyper“-application layer, serving as a unique and neutral interface on top of the today’s existing Layer-7-implementations.

Critics may arise from the fact that additional layers may consume additional processor resources and may result in even worse response time characteristics. All these constraints already today represent severe problems. For sure these remarks are very serious ones. On the other hand there will be more computer/CAN power being available with the progress steadily made in semiconductor technology. And the CAN community should be prepared to tell the semiconductor manufacturers what they would like to see integrated into hardware. Therefore these ideas must be pursued seriously now and great attempt for world wide acceptance, harmonization, standardization must be done now to be prepared for the future, to be prepared for the next future „Hardening Process of CAN System Software“.

1. Literature

- [1] International Standard ISO 11898 „Road vehicles - Interchange of digital information - Controller area network (CAN) for high speed communication“
ISO Reference number ISO 11898:1993(E), First edition 1993-11-15
- [2] International Standard ISO 11519-2 „Road vehicles - Low-speed serial data communication - Part 2: Low-speed controller area network (CAN)“
ISO Reference number ISO 11519-2:1994(E), First edition 1994-06-15
- [3] „CAN Specification Version 2.0“
1991 Philips Semiconductors Unternehmensbereich der Philips GmbH
Burchardstrasse 19, D-20095 Hamburg, Germany
- [4] W. Lawrenz et al.: "CAN Controller Area Network, Grundlagen und Praxis"
Hüthig Verlag, 1994, ISBN 3-7785-2263-9
- [5] K. Etschberger et al.: „CAN Controller-Area-Network, Grundlagen, Protokolle, Bausteine, Anwendungen“
Carl Hanser Verlag München Wien, 1994, ISBN 3-446-17596-2
- [6] CiA: CAN in Automation e.V. Users Group
Am Weichselgarten 25, D-91058 Erlangen, Germany
Tel.: +49 9131 601091, Fax.: +49 9131 601092
- [7] W. Lawrenz: "Network Application Layer"
SAE Detroit, February/March 1994, Paper 940141
- [8] W. Lawrenz: "Networked Systems: High Level Design and Test Philosophy and Tools"
SAE Detroit, February/March 1995, Paper 950296
- [9] "CAN-Tools Users Manual"
I+ME GmbH, Ferdinandstr. 15 A,
D-38300 Wolfenbüttel, Germany, Phone: +49 5331 97070
- [9] "System Builder Users Manual"
I+ME GmbH, Ferdinandstr. 15 A,
D-38300 Wolfenbüttel, Germany, Phone: +49 5331 97070
- [10] "CAL Users Manual"
I+ME GmbH, Ferdinandstr. 15 A,
D-38300 Wolfenbüttel, Germany, Phone: +49 5331 97070
- [11] "SDS"
HCB AG, Webergutstr. 4, CH-3052 Zollikofen, Switzerland
Tel.: +41 31 911 0763, Fax.: +41 31 911 6836
- [12] "DeviceNet"
Allen-Bradley, Düsseldorf Str. 15, D-42781 Haan, Germany
Tel.: +49 2104-960193, Fax.: +49 2104-960158
- [13] U. Kiencke et al.: "Open Systems and Interfaces for Distributed Electronics in Cars (OSEK)"
SAE Detroit, February/March 1995, Paper 950291
- [14] W. Lawrenz: "Vernetzte Systeme im Automobil der Zukunft - Modellbildung, Höhere Lagen, Werkzeuge."
Förderkennzeichen: TV 8933 Prometheus-Phase II, TIB Hannover, September 1993
- [15] C. E. Shannon: „Communication in the Presence of Noise“
Proc. IRE, No. 37, 1949, pp 10-21