1

The CAN Bus – general

COPYRIGHTED MATERIAL

A bus is always a bus – but there are ‘buses’ and ‘buses’!

In fact, from one bus to the next, the problems to be solved remain the same, but the different characteristics of the proposed fields of application modify the hierarchical order of the parameters to be considered, and result in the development of new concepts in order to find neat solutions to the difficulties encountered. Here is a quick list of the virtually unchanging problems encountered in bus and network applications:

- network access concepts, including, clearly, problems of conflict, arbitration and latency;
- deterministic or probabilistic aspects and their relationship with the concepts of real-time or event-triggered systems;
- the concept of network elasticity (‘scalability’);
- the security of the data carried, in other words, the strategy for managing errors including their detection, signalling and correction;
- questions of topology, length and bit rate;
- questions of physical media;
- radio-frequency pollution, etc.

1.1 Concepts of Bus Access and Arbitration

‘Distributed’ real-time control systems, based on an operating system located within a single processor, interconnected by a communication network with distributed processors, are currently providing a significant addition to ‘parallel’ systems.

In addition to the simple exchange of data, the processing must be synchronized, in other words, the execution must follow certain interrelated logical sequences. In these systems, the messages relating to synchronization are generally short. They can be created by any process or event in the system and must be simultaneously receiving, in order to maintain the coherence of parallel processing.

Multiplexed Networks for Embedded Systems: CAN, LIN, Flexray, Safe-by-Wire... D. Paret
© 2007 John Wiley & Sons, Ltd

All stations independently generate messages concerning their respective tasks at random (event-triggered) instants.

The transmission requests contend with each other for access to the bus, leading to latencies[1] which are variable rather than constant.

Let us now examine the different principles of arbitration, which are in the running.

1.1.1 CSMA/CD versus CSMA/CA

CSMA/CD

For historic reasons, the Carrier Sensor Multiple Access/Collision Detect (CSMA/CD) arbitration procedure was initially developed to resolve these conflicts.

With this system, when several stations try to access the bus simultaneously when it is at rest, a contention message is detected. The transmission is then halted and all the stations withdraw from the network. After a certain period, different for each station, each station again tries to access the network.

It is well known that these data transfer cancellations during contention theoretically also decrease the carrying capacity of the network. The network may even be totally blocked at peak traffic times, which is unacceptable when the network is to be used for what are known as 'real-time' applications.

CSMA/CA

In view of the above problems, another principle (one of several) was carefully investigated. This is known as Carrier Sensor Multiple Access/Collision Avoidance (CSMA/CA).

This device operates with a contention procedure not at the time of the attempt to access the bus, but at the level of the bit itself (bitwise contention – conflict management within the duration of the bit). This principle has been adopted in the CAN (controller area network) protocol, and its operation is described in detail in this part of the book.

In this case, bus access conflicts can be avoided by assigning a level of priority to each message carried.

In the case of contention, the message having the highest priority will always gain access to the bus. Clearly, the latency of the messages will then depend markedly on the priority levels chosen and assigned to each of them.

1.1.2 The problem of latency

In the overall design of a system, in order to take all parameters into account, the latency is generally defined as the time between the instant indicating the request for transmission and the actual start of the action generated by this.

For the time being, and for the sake of simplicity, I shall define the latency of a message (t_{lat}) as the time elapsing between the instant indicating a request for transmission and the actual start of the transmission.

---

[1] See Section 1.3.

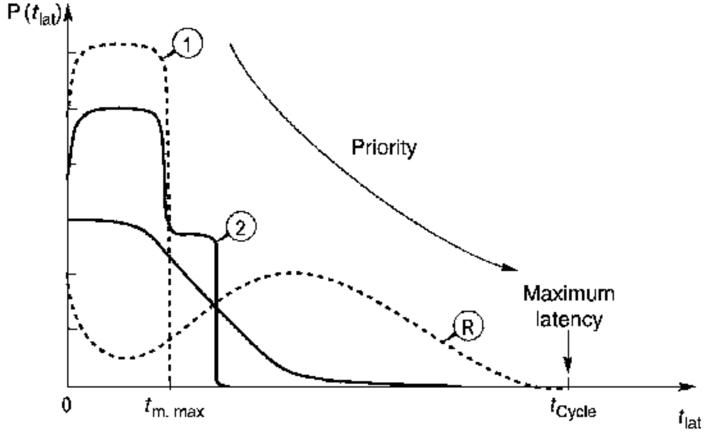


Figure 1.1

This concept is widespread and used in statistical analysis, mainly in what are known as 'real-time' systems. The reason is simple: only a few specific messages really need to have guaranteed latency, and then only during peak traffic times. We must therefore consider two kinds of messages:

- $R =$ messages whose latency must be guaranteed,
- $S =$ the rest,

and of course $M = R + S$, the total number of messages.

The curve in Figure 1.1 shows the probability distribution of the latency as a function of the latency, where the transmission request is made once only.

The specific value t_{cycle} is the time representing a mean activity cycle of the network consisting of M messages having a temporal length t containing N bits. The curves depend on the priority of the messages. The probability distribution of priority '1' returns to 0 immediately after the transfer of the longest message.

1.1.3 Bitwise contention

This CSMA/CA principle, used in the CAN bus (patented since 1975), of the same type as that of the I2C bus, introduces some constraints, both in the representation of the signal in the physical layer and in relation to the maximum geometry '1' of a network operating in this way.

Thus, during the arbitration phase, in order to have higher priority bits in the network erasing those of lower priority, the physical signal on the bus must be

- either dominant,[1] for example, the presence of power, current, light or electromagnetic radiation,
- or recessive,[1] for example, an absence of power.

---

[1] Increase your Word Power.

By definition, when a dominant bit and a recessive bit are transmitted simultaneously on the bus, the resulting state on the bus must be the dominant state.

1.1.4 Initial consequences relating to the bit rate and the length of the network

Now, here are a few words about the consequences of what I have described above (a part of Chapter 3 will also deal with this thorny question).

We know that the propagation velocity of electromagnetic waves v_{prop} is of the order of 200,000 km s^{-1} in wires and optical fibres (or, expressed another way, each wave takes approximately 5 ns to cover 1 m, or again travels at 200 m μs^{-1}).

Theoretically, in a system operating by bit contention, a bit can travel from one end of the network to the other before being detected on its arrival.

Now, it is possible that, a few 'micro-instants' before the bit reaches its destination, the other station, having seen nothing arrive at its terminal, may decide to start transmitting in the other direction. Head-on collision! Disaster looms!

If we call t_{bus} the time taken by the signal to travel the maximum length of the network, the global sum of the outward and return times for the propagation of the signals on the bus is

$$2t_{\text{bus}} = 2\frac{l}{v_{\text{prop}}}.$$

For example, if $l = 40$ m, then $t_{\text{bus}} = 200$ ns.

To enable the station which sent the initial bit to manage the conflicts, the necessary duration of the bit, known as the bit time or t_{bit}, must be longer than t_{bus}. And for the sake of completeness, we must allow for the time taken (or required) to sample and process the bit at the station where it arrives.

To evaluate the minimum bit time, $t_{\text{bit-min}}$, of the proposed network, it is necessary (Figure 1.2) to allow for

- the outward propagation delays, t_{out},
- the inward propagation delays, t_{in},

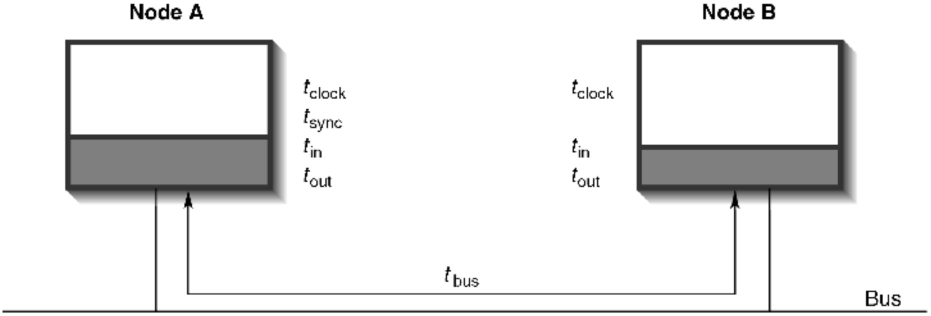


Figure 1.2

- the delays due to synchronization, t_{sync},
- the phase differences due to clock tolerances, t_{clock},

giving a total $t_{\text{bit-min}}$ of

$$t_{\text{bit}} = 2t_{\text{bus}} + 2t_{\text{out}} + 2t_{\text{in}} + t_{\text{sync}} + t_{\text{clock}}.$$

Example.
With a bit rate of 100 kbit s^{-1}, that is a bit time of 10 μs, we can achieve a network length of approximately 900 m.

All these concepts are described in detail in Chapter 3.

1.1.5 The concept of elasticity of a system

The architectural and topological configuration of a distributed system is generally different from one application to another, and, for a single application, it can also develop or be changed over a period, depending on the requirements to be met. To make things clearer, consider the example of the installation of industrial production lines, which, although they always have the same overall appearance, need to be reconfigured from time to time according to the products to be made. Where systems or networks are concerned, we generally use the term ‘elasticity’ to denote the capacity to withstand a change of configuration with the least possible amount of reprogramming in relation to the data transfer to be provided.

Let us look more closely at the problems associated with the elasticity of a network.

The information received and processed somewhere in a distributed system must be created and transmitted to a station. There is no logical alternative to this. There are two possible cases:

- New information is to be added. Any new information requires a new message transfer and therefore a reprogramming of the communication. In this case, the station which previously sent this specific information element must be reprogrammed according to the new one, whereas the other stations remain unchanged.
- A different situation occurs when a pre-existing specific information element has to be either transmitted from another station or received by additional stations. In this case, the additional receiving station must be reprogrammed to receive the information.

1.1.6 Implication of the elasticity of a system for the choice of addressing principle

Conventional addressing, generally consisting of a ‘source’ address on the one hand and a ‘destination’ address on the other hand, cannot provide a system with good structural elasticity. This is because any messages that have to be rerouted will require modifications, even if this is not logically necessary, as mentioned above.

For the CAN concept, in order to provide good elasticity in the system, it was decided to use another addressing principle, based not on the source and destination addresses but on the content of the message itself. This implies two things:

- A message has to be transmitted to all the other stations in the network (the term used for this principle is ‘broadcast diffusion’).

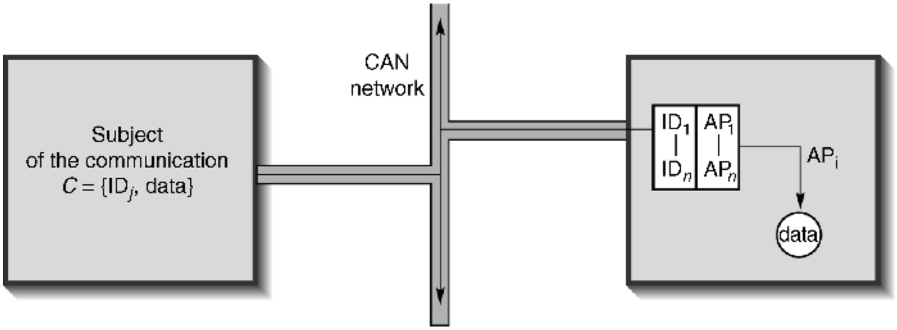


Figure 1.3

- The selection processing of the transmitted message is then carried out by what is called ‘acceptance filtering’ at each station.

For this purpose, the message is labelled with an identifier ID(i), which is then compared with the list of messages received (or to be received) at each station.

This list contains address pointers AP(i) towards the communication buffer, enabling the content of the message to be stored (Figure 1.3).

Therefore, *all* the messages are simultaneously received over *all* of the network, and the data consistency is thus guaranteed in distributed control systems.

1.2 Error Processing and Management

1.2.1 The concept of positive and negative acknowledgements

The elasticity of systems and identification based on the message content complicate the processing of errors when they occur.

An example of a conventional method of (non-)detection of errors is the return of what is called a ‘positive’ acknowledgement from the receiving station to the transmitting station, when a message is received correctly. The key information contained within such a positive acknowledgement is generally the address of the receiving station.

In the CAN concept, this idea of a local address completely disappears, and the identifier ‘labelling’ the message is transmitted to all the participants and received everywhere in the network. This makes it necessary to execute a local error processing task in each of the stations in the network, in the absence of such local message addresses. To achieve this, the CAN protocol concept uses a combination of positive and negative acknowledgements.

The positive acknowledgement ACK+ is expressed as follows:

$$\text{ACK+} = \text{ACK} + (i) \text{ for any } (i).$$

It is sent from all stations (i) which have received the message correctly and expresses this (positive) acknowledgement during a single precisely defined time interval (ACK time slot).

The principle described above can also be expressed in two basic forms:

- an optimistic view of the problem, in which we can say that the positive acknowledgement indicates that 'at least one station has received the transmitted message correctly';
- a rather less optimistic (but not totally pessimistic) view: 'The negative acknowledgement of a message must take a form such that it indicates (or will indicate) that there is at least one error in the global system'.

In the latter case, the message indicating the presence of the error must be sent over the network immediately after the detection of the error. This method will ensure that the system can be resynchronized immediately within a single message frame. The latter point is crucial for the security of the applications considered.

1.2.2 Error management

The combination, the redundancy and the analysis of the positive and negative acknowledgements sent from the error processing devices of the different stations are exploited to provide strong presumptions concerning the source of an error (either from the transmitting station or from one of the receiving stations), for example

- The presence of at least one positive acknowledgement sent from a receiver, combined with an error message, signifies that the message has been transmitted correctly at least.
- Conversely, the absence of a positive acknowledgement, together with an error message, indicates that all the receiving stations have detected an error and that there is a strong presumption (or probability) that the error is located in the transmitting station.

1.2.3 Error messages

The error messages used for the CAN concept are of two types (Figure 1.4).

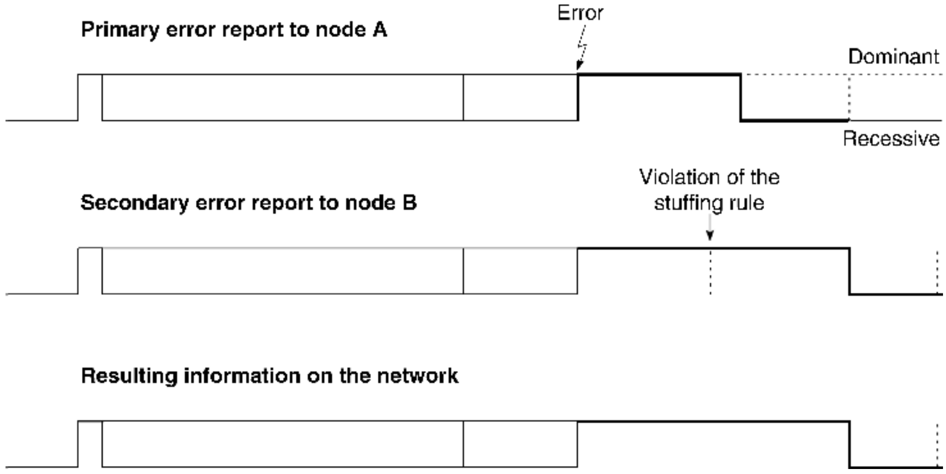


Figure 1.4

Primary error report

A station (or several stations at once) detect(s) an error, causing it (or them) to immediately transmit an error message.

At this point, the error message consists of a sequence of dominant bits and the current message is aborted (not fully processed).

Secondary error report

Following this cancellation of the message, all the other stations detecting a violation of the format rules transmit an error message on their own behalf, these messages being combined with the first one and thus increasing its duration.

Primary error reports are more probable for a station where the error has occurred than secondary error reports, which are generated in response to the aforementioned error.

1.2.4 The concept of an error management strategy

To manage errors correctly, the ultimate aim is to develop a strategy for the processing of errors. The quality of the processing will depend markedly on whether or not the strategy defined for a given field of applications is proactive.

For the CAN protocol, a strategy called an error (or fault) ‘confinement’ strategy, shown in outline in graphic form in Figures 1.5 and 1.6, has been defined (and is described in much greater detail in Chapter 2). Here is a brief summary.

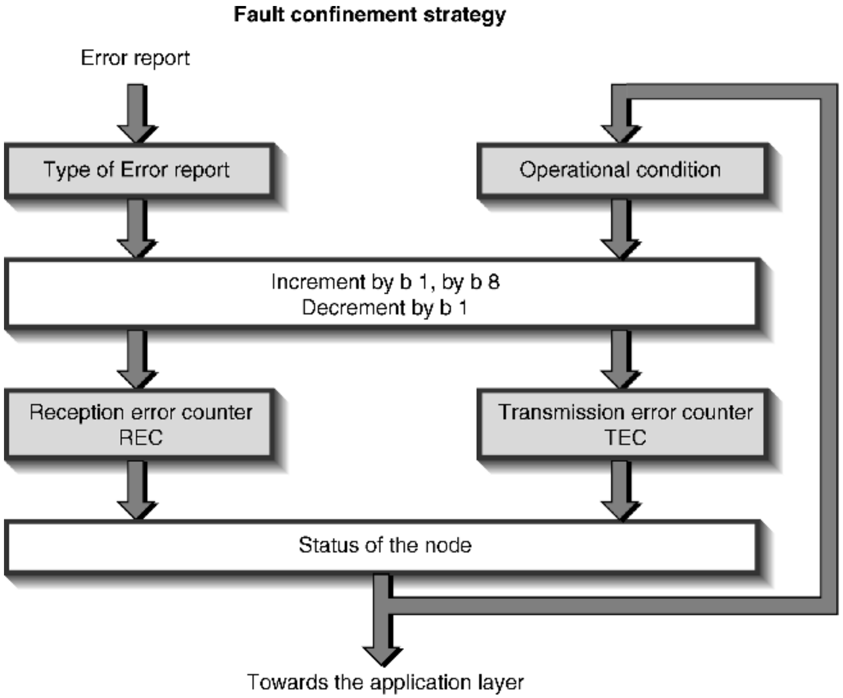


Figure 1.5

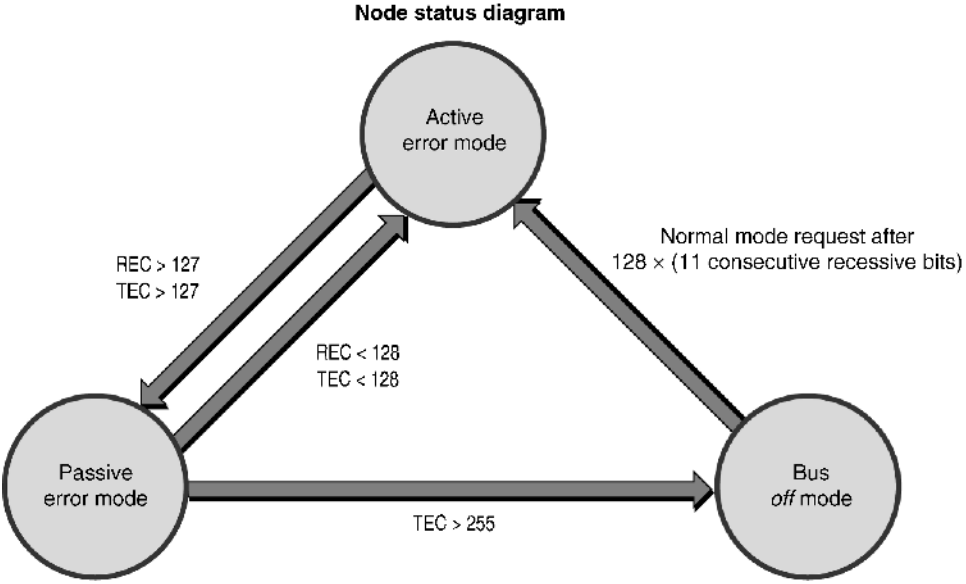


Figure 1.6

First, each station has to have two separate error counters: one to record what happens during the transmission of a message and the other to execute a similar task during reception.

Then, according to the type of error report and the operating conditions of the station at the instant in question, the counters are incremented with different weightings according to certain conditions or are decremented.

The purpose of these counters is to record information originating directly or indirectly from all the other stations. Using this device, the counters carry out an averaging operation, giving an approximation of the statistical values of the local quality of the network at a given instant.

To provide more support for this strategy, it was also specified that if too many errors were attributed by such statistical means to a specific station, its state would be transferred from what is called an ‘active error’ mode to a ‘passive error’ state, in which it would no longer communicate but would always be active in the management of errors which may occur on the network.

If there are too many transmission errors in the network, the network could be blocked, making all traffic impossible.

To avoid this, it is necessary to specify that, above a certain limit (fixed at 255 for CAN), the station switches to another new state, called *bus off*, during which the station appears to be physically disconnected from the bus so as not to block the bus for any reason.

In this case, its input stage stays in a high impedance state, thus making it possible to observe the signal travelling in the network (monitor function).

This ends the summary of the key points for the study of this new protocol concept, together with the major ideas used to resolve the problems generally raised by the applications.

At this stage, you have the necessary background to study the actual operation of this protocol and of the associated bus.

Now for the work! The bus is waiting, so let us go... (see also *The I2C Bus* [same author, same publisher] in which, at that time, we were only waiting for the... bus!).

1.3 Increase Your Word Power

To ensure that we are using the same vocabulary, I now offer for your guidance a few unillustrated extracts from dictionaries:

- *Avoidance*: the fact of avoiding, from the verb ‘to avoid’.
- *Confinement*: the act of confining (keeping within confines, limits, edges).
- *Consistency*: keeping together, solidity.
- *Contention*: argument, dispute, from the Latin *contentio* (struggle, fight).
- *Identifier*: ‘that which identifies’.
- *Latent*: in a state representing latency (see ‘latency’).
- *Latency*: time elapsing between a stimulus and the reaction to the stimulus.
- *Recessive*: persisting (still active) in the latent state.

1.4 From Concept to Reality

The foundations of the CAN concept were laid, at a given period, on the basis of a state of the market and technology, with an attempt to extrapolate future trends, in respect of the systems and the techniques and technologies to be implemented to make them a reality.

The question that you may well ask is: Why CAN and not another protocol?

So here are some further explanations which will help you to understand; let us begin with a few words about ‘site buses’.

1.4.1 The site bus market

For a long time, many companies have been obliged to develop and suggest their own solutions for resolving substantially similar (or related) problems raised by links and communications between systems.

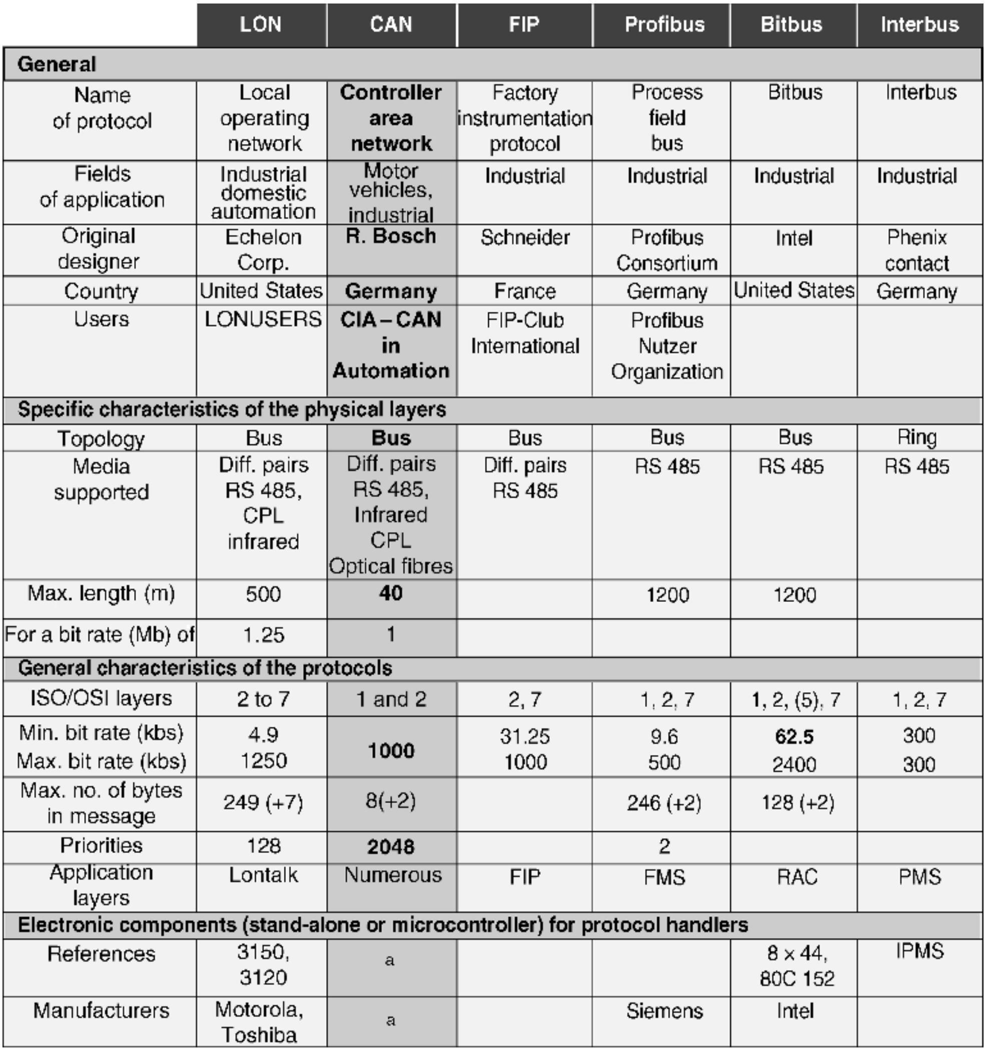
Nearly all of these solutions are known as ‘proprietary’, and because of the different interests of each of the companies involved, this has led to a significant fragmentation of the market. This diversification of the solutions and the small or medium quantities covered by each of them have therefore, unfortunately, led to a decrease in the quantities of specific components to be developed and produced in order to create a standard on this basis. Among these solutions, many serial buses and links have existed for a long time to resolve similar or virtually similar applications. Regardless of the markets supported (industrial or motor vehicle), the best known names (listed in alphabetical order to avoid offending certain sensitivities) are Batibus, Bitbus, EIB, FIP bus, J1850, LONwork, Profibus, VAN, etc.

Figure 1.7 provides a summary of the numerous buses for covering some or all of these applications.

So why use CAN, which is also a proprietary system?

1.4.2 Introduction to CAN

The size of the motor vehicle market (several million units per year) has led component manufacturers to design integrated circuits handling the CAN protocol. It has also



	LON	CAN	FIP	Profibus	Bitbus	Interbus
General						
Name of protocol	Local operating network	**Controller area network**	Factory instrumentation protocol	Process field bus	Bitbus	Interbus
Fields of application	Industrial domestic automation	Motor vehicles, industrial	Industrial	Industrial	Industrial	Industrial
Original designer	Echelon Corp.	**R. Bosch**	Schneider	Profibus Consortium	Intel	Phenix contact
Country	United States	**Germany**	France	Germany	United States	Germany
Users	LONUSERS	**CIA – CAN in Automation**	FIP-Club International	Profibus Nutzer Organization		
Specific characteristics of the physical layers						
Topology	Bus	**Bus**	Bus	Bus	Bus	Ring
Media supported	Diff. pairs RS 485, CPL infrared	Diff. pairs RS 485, Infrared CPL Optical fibres	Diff. pairs RS 485	RS 485	RS 485	RS 485
Max. length (m)	500	**40**		1200	1200	
For a bit rate (Mb) of	1.25	1				
General characteristics of the protocols						
ISO/OSI layers	2 to 7	1 and 2	2, 7	1, 2, 7	1, 2, (5), 7	1, 2, 7
Min. bit rate (kbs) Max. bit rate (kbs)	4.9 1250	**1000**	31.25 1000	9.6 500	**62.5** 2400	300 300
Max. no. of bytes in message	249 (+7)	8(+2)		246 (+2)	128 (+2)	
Priorities	128	**2048**		2		
Application layers	Lontalk	Numerous	FIP	FMS	RAC	PMS
Electronic components (stand-alone or microcontroller) for protocol handlers						
References	3150, 3120	[a]			8 × 44, 80C 152	IPMS
Manufacturers	Motorola, Toshiba	[a]		Siemens	Intel	

[a] The list is too long to be shown here.

Figure 1.7

succeeded in reducing costs significantly, which is not necessarily the case with other buses, which are frequently restricted to smaller scale applications without the benefit of dedicated components or keen prices. Manufacturers have therefore considered the arrival of the CAN bus from another viewpoint (the very special viewpoint of the performance/cost ratio) and have suddenly decided that this type of bus fully satisfies them.

It is surprising what can be achieved by taking a fresh look!

1.4.3 The CAN offer: a complete solution

The strength of the CAN concept, its promotion and its success, lies in the motivation of the people involved in the project to provide the users, at just the right time, with everything they need and cannot easily find elsewhere to deal with the same problems.

From considerable personal experience, I know that this requires a lot of work, but it always brings results when it is well organized. Indeed, all the components for the intelligent development of CAN products have appeared and become available within 2 or 3 years (a period well suited to the market); these components are

- a precise and complete protocol, spelt out clearly;
- the ISO standards for motor vehicle applications;
- competing families of electronic components;
- development of awareness in the industrial market;
- technical literature (articles, books, etc.);
- conferences and congresses for increasing awareness, training, etc.;
- formation of manufacturing groups (CiA, etc.);
- supplementary recommendations for the industry, concerning for example the sockets (CiA) and the application layers (CiA, Honeywell, Allen Bradley, etc.);
- tools for demonstration, evaluation, component and network development, etc.

In short, it all helps!

And now, before we move on to purely technical matters and the applications of the CAN bus, here by way of light relief are a few lines about the history of the protocol.

1.5 Historical Context of CAN

A little history always helps us to understand the present, so let us spend some time on this.

Since the early 1980s, many electronic systems have appeared in the car industry. Essentially, this has taken place in three major steps:

- the era when each system was completely independent of the others ... *and everyone lived his own life*;
- a period in which some systems began to communicate with each other ... *and had good neighbourly relations*;
- finally, our own era when everyone has to ‘chat’ with everyone else, in real time ... ‘*think global*’, *the world is a big village*.

But let us go back to the beginning.

In those days ... at the start of the 1980s, we had to think about managing future developments. It was in this spirit that, shortly after the appearance of the I2C and D2B ‘serial type’ buses in the market, many companies concerned with industrial applications (public works, etc.) and some major car manufacturers became interested in communication systems operating (almost) in real time between different microcontrollers, especially for engine control, automatic transmission and antiskid braking systems. For several years, these companies tried to fill the gap by attempting to combine I2C with D2B, in the absence of dedicated solutions.

This was initially done with the aid of conventional UARTs, like those found in ordinary microcontrollers available in the market. Sadly, it soon became clear that although these had very useful properties, they could only reach one part (mainly the passenger area of the vehicle) of the target (the whole vehicle). This form of communication management provided no support, or very poor support, for multimaster communications, and other devices had to be devised and developed. Moreover, their maximum bit rate and the security of the information carried were inadequate. Consequently, there was a gap due to the absence of a bus capable of providing 'fast' multimaster communications, operating over a 'correct' distance and 'insensitive' to its carrier.

In 1983, on the basis of certain performance levels achieved by the I2C asymmetric bus (bitwise arbitration), the D2B symmetric bus (differential pair) and many other factors, the leading German motor components company R. Bosch GmbH took the decision to develop a communication protocol orientated towards 'distributed systems', operating in real time and meeting all the company's requirements. This was the start of the development of CAN. It would be a great untruth to claim that the sole intention was to keep this system under wraps and not use it, when the company was one of the world leaders in motor components.

At this point in the story, let me emphasize the first significant point. This is that the management of R. Bosch GmbH decided to become directly involved in a major way, by cutting through hierarchical relationships and stimulating the development, with the result that the major customers of Bosch were made aware of the state of progress of the project at the start of 1984.

The second major point is that a motor component manufacturer, howsoever large, cannot achieve anything with a concept of this type unless it forms a partnership with universities (particularly the Fachhochscule Braunschweig at Wolfenbüttel, where Professor Wolfhard Lawrenz was later to suggest the name 'controller area network or CAN') and with the creators of known integrated circuits, to make these 'silicon dreams' a reality. Each has its own part to play. These partnerships were established in 1985 with the American giant Intel (useful for ensuring the vital subsequent promotion in the United States) and then, some time later, with Philips SemiConductors, to fill out the European side of the picture. Of course, since that time, many component manufacturers have followed in their footsteps (Siemens, later Infineon, Motorola, later Freescale, NS, TI, MHS, later Temic then Atmel, most of the Far Eastern producers, etc.). In short, the forces were massing.

In the spring of 1986, it was finally time to reveal the new system to the world. The first presentation about CAN was made exclusively to members of the well-known SAE (Society of Automotive Engineers). Where was this? Where else but in the United States, where the car is king, and in a very significant city, Detroit, the cradle and stronghold of the American automobile: if this showed some bias towards the United States (mainly among the car manufacturers, the SAE Trucks and Bus Committee), the Europeans would soon become interested.

Following this, in 1986, everyone converged on the ISO, asking them to. .. set the standards (what else would ISO do?). Any practitioner at the ISO will tell you that the standardization of a protocol requires many years, but forceps can also be useful in helping with the delivery of the little newcomer!

Finally, in the middle of 1987, the reality took shape in the form of the first functional chips, and in 1991 a first top-range vehicle (German) rolled off the production line, complete with five electronic control units (ECUs) and a CAN bus operating at 500 kbit s^{-1}.

During the same period, the ‘internal’ promotions (for motor applications) and ‘external’ promotions (for industrial applications) were created and actively supported by the SAE and OSEK for the motor industry and also by CAN in Automation (CiA – a group of industrialists, mainly German but subsequently international) for other fields.

So it was the ‘engine’ of the motor industry that brought this concept to light, but its typically industrial application is not limited to this market. Without ignoring these first applications, therefore, I will try to counteract the general view that CAN is a protocol designed purely for the motor industry, by showing you that it is a highly efficient system for fast local networks.

1.5.1 CAN is 20 years old!

By way of documentation, the table below shows the main stages of development of CAN during its first 20 years of life.

1983	Start of development of CAN at R. Bosch GmbH.
1985	V 1.0 specifications of CAN. First relationships established between Bosch and circuit producers.
1986	Start of standardization work at ISO.
1987	Introduction of the first prototype of a CAN-integrated circuit.
1989	Start of the first industrial applications.
1991	Specifications of the extended CAN 2.0 protocol: • part 2.0A – 11-bit identifier; • part 2.0B – 29-bit identifier. The first vehicle – Mercedes class S – fitted with five units communicating at 500 kbit s^{-1}.
1992	Creation of the CiA (CAN in Automation) user group.
1993	Creation of the OSEK group. Appearance of the first application layer – CAL – of CiA.
1994	The first standardization at ISO, called *high and* *PSA (Peugeot Citroën) low speed*, is completed. and Renault join OSEK.
1995	Task force in the United States with the SAE.
1996	CAN is applied in most ‘engine control systems’ of top-range European vehicles. Numerous participants in OSEK
1997	All the major chip producers offer families of CAN components. The CiA group has 300 member companies.
1998	New set of ISO standards relating to CAN (diagnostics, conformity, etc.).
1999	Development phase of time-triggered CAN (TTCAN) networks.
2000	Explosion of CAN-linked equipment in all motor vehicle and industrial applications.
2001	Industrial introduction of real-time time-triggered CAN (TTCAN) networks.
2003	Even the Americans and Japanese use CAN!
2008	Annual world production forecast: approximately 65–67 million vehicles, with 10–15 CAN nodes per vehicle on average. Do the sums!

1.5.2 The CAN concept in a few words

It was decided from the outset that CAN should be capable of covering all communication applications found in motor vehicles, in other words, it should carry and multiplex many types of messages, from the fastest to the slowest.

Because of its origin in the car industry, CAN was designed to operate in environments subject to a high level of pollution, primarily due to electromagnetic disturbance. In addition to the transmission reliability provided by an efficient error detection mechanism, CAN has multimaster functionality to increase the possibility of providing fast recovery from errors after their detection. Also, in the case of bus access conflicts, the bitwise arbitration principle adopted makes it possible to provide a 'non-destructive' arbitration method when several stations attempt to send a message simultaneously. This means that, in a contention situation, one station will always gain access to the bus (namely the station sending the highest priority message) and will then finish the communication on its own. Because of this method, none of the bus communication capacity (the bus bandwidth) is lost in the management of bus access conflicts.

The use of this type of non-destructive arbitration and hierarchically ranked messages also makes it a simple matter to meet, in real time, the response times required for the control of systems in which the bus bit rates are not very high.

The disadvantage of this bitwise arbitration method lies in the fact that the maximum length of the network is tied to the chosen bit rate, a high bit rate corresponding to a short-distance network.

In principle, in order to minimize the electromagnetic noise which is mainly due to fast edges of electronic signals during communication on the bus, the communication bit rate should be as low as possible.

See Chapter 2 on the CAN protocol for more details.

1.5.3 The market for CAN

In parallel with the advance of motor applications, a large number of industrial applications – to be fully discussed in this book – have emerged. This success is largely due to the rapid appearance in the market of inexpensive electronic components (integrated circuits) for managing the communication protocol (because the CAN protocol only covers layers 1 and 2 of the ISO/OSI model).

By way of a summary, in the middle of 2004, Figure 1.8 shows the cumulated total of nodes in the market and a comparison with other industrial solutions.

The motor vehicle market

The motor vehicle market is most important in terms of quantity, simply because of the annual volume of nodes implemented per vehicle produced. Moreover, in future, at least in Europe, the number of CAN nodes on each vehicle will be approximately 5–10 for the engine system of the vehicle, about 10 for the body part, and finally 15, 20, 25 or more for the passenger compartment, depending on the level of equipment and comfort features fitted to vehicles.

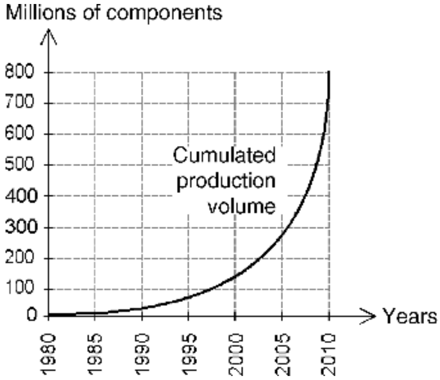


Figure 1.8

The industrial automation market

Although the relative volume of this market is smaller than that described above, it still represents an important amount in absolute terms. Even so, it should be noted that this market was the first to be established on an industrial scale, because of the short lead times required by small and medium businesses to respond to market demand (the development times for industrial applications generally vary from 6 to 12 months, whereas in the motor industry they are of the order of 2–3 years). In 1996, for the last time, the quantity of nodes produced for the automation market exceeded the number for the motor industry market.

Chapters 4 and 5 will describe the wide field of industrial applications which have given strength and life to this concept.

Application layers and support organizations

Today, for all these markets, there is only one communication protocol: CAN. Regarding the application layers (see the book *Le bus CAN – Applications* [*The CAN Bus – Applications*]), there are many competing proposals, adapted to different application criteria. To gain a clearer idea, let us now examine a list of the main names.

For industrial applications, the principal ones are

- CAL, produced by CAN in Automation,
- CANopen, produced by CAN in Automation,
- DeviceNet, produced by Allen Bradley–Rockwell,
- SDS (smart distributed systems), produced by Honeywell,
- CAN Kingdom, produced by Kvaser,

and, for motor vehicle applications:

- OSEK/VDX, produced by OSEK (open systems and interfaces for distributed electronics in car group),
- J 1939, produced by SAE.

It should also be noted that many development, promotion and standardization organizations (such as CiA – CAN in Automation, ODVA, OSEK, etc.) have sprung up during the same period.

Finally, when a system is devised, it is quite normal for one or more similar competing systems to appear. This is just what happened with the advent of many similar types of bus in the United States, Japan and France (for example, the VAN bus in France, which was devised several years ago by PSA but which had its industrial development halted in 2004). This has been a brief summary, in an intentionally lightweight form, of the historical context of this bus (which is actually a very serious subject). In reading this you would have understood how certain joint technical and marketing approaches can sometimes make a concept into an industrial success story.

1.6 Patents, Licences and Certification

To conclude this chapter, a few words in passing concerning patents, licences and standards are offered to you by way of light relief, before the lengthy description of the CAN protocol and its numerous physical implementations.

1.6.1 Documents and standards

Now we come to the matter of standardization: We must comply with certain terms and be accurate in their definition.

The original document

The original CAN protocol is described in a document issued by R. Bosch GmbH (the latest edition is dated 1992) and as such it is not an international standard. This document describes the protocol without consideration of all the lower physical layers and all the higher level software application layers. It covers layer 2 of the OSI (Open Systems Interconnect) model in full (LLC and MAC) and part of layer 1 (PLS), giving free rein to all possible applications. On its appearance, one of the aims of this document was to specify layers 1 and 2 of the OSI model without delay, to enable the chip producers to create suitable electronic components as soon as possible to handle the protocol, without prejudicing future applications and while leaving the way open to many new fields.

ISO standardization

Over the years, the original CAN documents were filled out and submitted to the ISO so that official international standards could be used as a reference by all those who wished to adopt this protocol. These documents have also attempted to comply where possible with the structure of the OSI communication model. At present (2007), the principal standards relate to motor vehicle applications and are known by the following references:

- ISO 11898-x – road vehicles – interchange of digital information. This is the cornerstone of the CAN standard, consisting of five documents:

  - ISO 11898-1 (*data link layer and physical signalling*);
  - ISO 11898-2 (*high-speed medium access unit*);
  - ISO 11898-3 (*low-speed fault-tolerant medium-dependent interface*);
  - ISO 11898-4 (*time-triggered CAN*);
  - ISO 11898-5 relates to high-speed CAN and low-power applications.

The highly detailed documents of parts 1 and 2 faithfully reproduce all the text proposed by Bosch (LLC, MAC, PLS), in other words, the former ISO 11898 standard dating from 1993, while supplementing it with much other material mainly concerning the 'media' part of the physical layer (PMA), not included in the original document. The same applies to part 3, which describes certain forms of bus dysfunction in relation to the media recommended in the documents (wires short-circuited, joined to the power supply, wires cut, etc.).

- ISO 11519-x (note that this standard has been superseded by ISO 11898-3):
  - ISO 11519-1 – general and definitions – road vehicles – low-speed serial data communications;
  - ISO 11519-2 – low-speed CAN – road vehicles – low-speed serial data communications.

Many other non-ISO documents are associated with these, especially in relation to the application layers such as CiA, CANopen, SDS, DeviceNet, etc., and in relation to the specific connectivity.

This global set of documents thus enables all designers to develop marketable CAN products, at least as regards the electronic operation of the system (but note that many other minor points of detail are also specified in these documents).

Although they are ISO standards, these documents give no information on the types of connectors recommended or on the higher level application layers. You may say that this is not their role, and you will be (partially) correct. Indeed, in the absence of any law or formal implementation order, anyone is free to obey or disregard the standard. Moreover, regardless of the existence of these standards, car manufacturers are entirely free to do what they want in their vehicles. Clearly, it must be recognized that it is more practical to have a common standard if we wish to use parts from different equipment and component manufacturers. This is another story, largely outside the remit of this book. Briefly, these standards form a good base for future possibilities of interconnection with products from different sources.

For information, no official document mentions the terms 'full CAN' and 'basic CAN' – this should be remembered.

The CiA

Another event occurred when many companies became interested in using the CAN protocol and wanted to create industrial applications. These companies (major industrial groupings or small or medium businesses) wished to use the bus but found themselves short of resources when times were hard and, realizing that the future lay with CAN, decided to form a group in order to fill some of the gaps in the documents cited above and ensure that they had all the necessary elements for communication. Thus, the CiA (CAN in Automation – international users and manufacturers group) was set up in March 1992, with the following mission (quote): ‘to provide technical, product and marketing intelligence information in order to enhance the prestige, improve the continuity and provide a path for advancement of the CAN network’.

First, this group is international rather than purely German, an enormous advantage for the promotion of the protocol. Second, it includes a large number of companies (more than 500) from all sectors, which complement each other and which have the aim of producing CAN products with very good possibilities of interoperation.

In short, it was a priority for all these members to specify all the physical interfaces (cables, connectors, screens, etc.) and higher level application software (level 7 of the OSI/ISO layers). Like a ‘little ISO’, the participants created technical committees and working groups, leading to the issue of a set of ‘CiA recommendations’, filling practically all the blank areas left by the ISO.

These CiA recommendations are called CiA draft standards (CiA DS xxx) for the physical part and CAN application layers (CAL) for the software layers. Other more specialized groups have acted in the same way regarding the high-level application levels orientated towards certain fields of application. Examples are Allen Bradley and Honeywell, which have developed the DeviceNet and SDS layers, respectively, for industrial applications (see Chapter 5).

1.6.2 Patents

Many patents have been filed since the development of the CAN protocol and its physical implementations. Some of these have led to epic arguments about their priority, but everything has now been settled. Patents are only one of the external aspects of industrial property protection and are generally only used to make some money, sooner or later, out of the conditions of licence use by friends and competitors in our splendid profession. This has now been done...

The sacrosanct rule of the ISO is that when a standard is officially published – as in the case of CAN – this does not in any way affect the licences and royalties payable to those entitled to them, but these royalties must be ‘reasonable’ and the licences must be ‘non-discriminatory’, as expressed by the acronym RAND (reasonable and non-discriminatory). I do not intend to detail the licence fees or royalty payments here, but I can tell you that they are not negligible. You should also know that, as ordinary users, you will certainly not be affected by this. Also, as with many other protocols, it is practically always necessary to use specific integrated circuits to handle CAN. Once again, therefore, it is the component manufacturers (chip producers) who are mainly concerned with licensing problems, not the end users.

1.6.3 Problems of conformity

It is all well and good to have components available for which the manufacturers have paid their licence fees, but one question remains: How can we know if the circuits offered in the market really conform to the standards?

In some situations, this can be a very important question. Consider, for example, the exact moment when you need to brake your car, if the ABS is controlled via the CAN bus. Suppose just for a moment that a bit is not in its proper place at that instant: Who is liable for the accident that may occur as a result? Manufacturers, equipment makers, component producers, the standard?

This underlines the crucial problem of the conformity of protocol handlers, line driver interfaces, etc. with the protocol. For some protocols, there are independent organizations (laboratories) charged with certifying that the product concerned can perform its function or with establishing conformity, approval, a label, etc.

At the present time, as regards the CAN protocol, this type of procedure exists in the form of a document, *CAN Conformance Testing*, reference ISO 16845, which is actually a specific application of the ISO 9646-1 test plan. Before this document existed, Bosch supplied what is commonly called a 'reference model' to its licence holders to overcome the problem. This is a software model providing a battery of tests to be conducted on a protocol handler. If the protocol handler responds correctly to all these tests, it will be considered as conforming to the protocol. This also means that, having passed these tests, the different implementations of protocol handlers will also be able to communicate correctly with each other.

The manufacturer of protocol handlers is therefore responsible for assuring himself, and then ascertaining via an independent certification organization that his product actually passes all the tests proposed by the reference model and the ISO 16845 standard . . . and the users are responsible for making it operate correctly!

1.6.4 Certification

A system can only operate correctly if it is consistent, in other words, if it has a real uniformity of operation. For example, in the present case, all the nodes of a network must react in the same way in normal operation to the same problem, the same fault, etc. There are two possible solutions: either to be the sole owner of a proprietary solution, keeping everything under control by a licensing system, or, for simple market reasons (availability of components, wider range of components, competitive pricing, etc.), to open the solution up to a standard, preferably ISO.

To maintain the technical integrity of the solution, it is a common practice to propose a standard describing in detail the functionality that is to be applied and that is to be provided, but leaving the concrete physical and electronic implementation to the choice of each user. Once a standard has been written, voted on, commented on, corrected and finally issued, we can start work . . . but sometimes the standard may simply be shelved. In principle, compliance with a standard is not compulsory. If so inclined, the government can stipulate its use by passing a law or an order. The same applies to a car manufacturer, who can decide to accept electronic equipment only if it conforms to this or that standard.

But who says that it conforms? This is the sensitive area of the application of a standard! Anyone can say that his product conforms to a standard, if he has made measurements, using his own equipment, to ensure that the values recommended by the standard are matched: but his word is only binding on himself. Unfortunately, this situation, although highly commendable and justifiable, does not satisfy everybody. This is because, on the one hand, the standard is not always very clear and may be subject to different interpretations, and, on the other hand, it is sometimes difficult to measure certain parameters in a reproducible way by the measurement methods in use. To avoid these problems and the ensuing lengthy arguments, it is a very common practice to write a second standard (!), entitled ‘Conformance Tests’, describing in detail the measurement methods and instruments used for the standard in question. Finally, it is also possible to certify independent organizations for making these measurements so that they can deliver a document (declaration, certificate, approval report, authentication report, etc.) certifying the results of these famous conformity measurements.

To return to the subject of CAN, Bosch was for a long time the only organization capable of giving its support in matters concerning protocol handling, based on its reference model mentioned above. Where the lower ‘high speed’ layer was concerned, everyone relied on the data sheets of different component manufacturers.

The problem became more critical with the appearance of the LS FT CAN (low-speed fault-tolerant CAN) physical layer. This is because in this case, as described in detail in Chapter 4, all the nodes of the network must signal the same type of fault at the same instant. In view of this, the GIFT (generalized interoperable fault-tolerant CAN transceiver) working group was set up. This group subsequently transferred its work to the ISO, participating strongly in the development of the ISO 11898-3 standard which describes the functional part of LS FT CAN, raising the problems of certification described above. As it takes a long time to convert a document to an ISO standard (of the order of 36–48 months), the GIFT group decided, via a special ICT (International Transceiver Conformance Test) committee, to establish a conformance test document itself, containing more than 100 specific tests, in order to obtain professional validity for this practice. This group also authorized the independent company C&S, managed by Wolfhard Lawrenz and already closely involved in the GIFT and ICT, to conduct these tests.

I will now proceed to examine the more technical matters and explain the operation of the CAN protocol.