By Rob Hulsebos,
System Engineer,
Philips Industrial Automation Systems

CAN/OPEN is Tackling the Competition with Interbus/S and LON

CAN is a protocol that describes a network for controlling I/O. This article explains CAN/OPEN profiles and their use in industrial applications. It also compares CAN/OPEN with other fieldbuses such as Profibus/DP and Interbus/S, and explores the future of industrial communication protocols.

Up to now, it was quite a job to build a CAN network with equipment from several suppliers. Even if all modules follow the same protocol, there is still some programming work to be done in the user's application. By using CAN/Open, it must be possible to build a system in an easier manner.

WHAT IS CAN?

Bosch originally designed controller Area Network for automation applications in vehicles (cars and lorries). CAN specifies the protocol for a network that is intended to control I/O. It is precisely because CAN is very resistant to electrically "contaminated" environments that it is so suited for industrial applications. The first initiative in using CAN therefore came from industry. In the meantime, CAN has been increasingly used in the automotive industry (see figure 1). The enormous sales of CAN chips to this branch means that the price of controllers is still falling. No single other fieldbus has such an advantage; even with the popular Interbus/S, the millionth chip was not supplied until mid-1996.

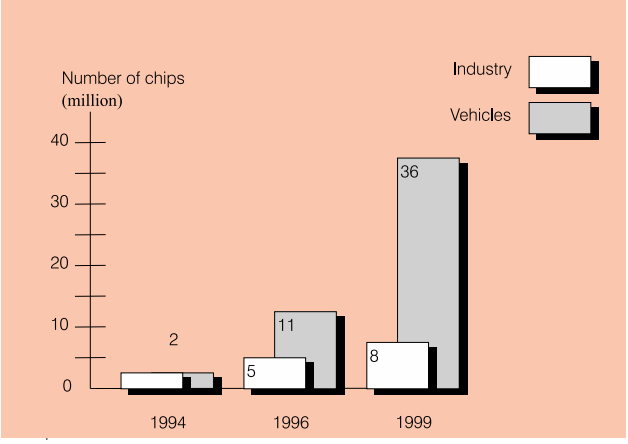

Figure 1. The first initiative in using CAN therefore came from industry.

As a result of the influence of the American car manufacturers, an extension to CAN was proposed a few years ago. We therefore now have "CAN 2.0A" and "CAN 2.0B", which differ from each other in a number of important points. This must be taken into account with the development (or purchase) of hardware. For application in industry, CAN 2.0A is mostly used. Another difference between CAN controllers is the possibility of being able to cope with "BasicCAN" or "FullCAN". There are mainly marketing terms; no difference can be seen on the network.

DEVELOPMENT OF SOFTWARE

CAN itself only specify the operation of the network and which messages can be exchanged. This is the function of OSI-layer 2. There is therefore no specification of the required functions in the remaining OSI layers, each user can fill this in himself.

This therefore has gone exactly so; many a supplier has filled all the holes himself. As could be expected, this has led to far-reaching incompatibilities between hardware and software from different suppliers, which is still one of the most important disadvantages of CAN in comparison with other fieldbus systems. We now have a whole family of different protocols (in alphabetic order):

CAL	The "CAN Application Layer" was originally developed by Philips Medical Systems, and later standardized by the CiA (CAN in Automation). It is regarded as the standard protocol for new applications. CAN/Open uses CAL as underlying protocol.
CAN-11	A protocol developed by BMW.
DeviceNet	Developed by Allen-Bradley (USA), now taken over by ODVA (Open DeviceNet Vendors Association). Approx. 130 suppliers support it.
Kingdom	Developed by Kvaser (Sweden), mainly for use in textile machinery.
LBS	The "Landwirtschaftliches Bus System" (DIN 9684) uses CAN in tractors and other agricultural equipment. The developments on LBS will continue for the coming years, and then be internationalized (ISO 11783).
M3S	The "Multiple Master, Multiple Slave" protocol is the result of a European research project with the purpose of

	developing a controller for wheelchairs.
MiCAN	The "Maritime Instrumentation CAN" protocol has been developed by the Norwegian company Read Matre Instruments for the control of PLCs and sensors in ships.
OSEK	The European car manufacturers have started the OSEK-project in 1995 (Offene Systeme fur die Elektronik im Kraftfahrzeug, Open Systems for the Electronics in Vehicles). They aim to develop an application protocol independent of the type of network being used.
SDS	Developed by Honeywell (Switzerland) for use in their sensors and actuators. It strongly resembles AS-i (Actuator-Sensor Interface). Approx. 40 suppliers support it.
SeleCAN	Developed by door Selectron (Swit-zerland) for their I/O-modules. It is also supported by Philips, Festo, Bernstein and Gespac. The protocol is very simple and fairly quick to implement. As a result, very fast I/O modules can be made.
VOLCANO	Has been developed by NRTT (UK) for automotive applications.

Naturally there are still enough companies who have developed their own protocols. The list above, however, gives only the most known or widespread protocols. Finally there are also CAN-products using the name CAN, such as TouCAN and PeliCAN. They are CAN-chips from Motorola and Philips.

STANDARDIZATION OF SOFTWARE

The CiA user's group was established fairly early on in the development of CAN. For the word 'users', one must mainly read: suppliers of equipment. But this is the case with many other user groups.

For a viable market, it is absolutely necessary that there is more standardization in software. This has resulted in the creation of CAL. It is a protocol that is situated on OSI-later 7, and therefore offers services to an application program. The first steps on the CAL path were made by Philips Medical Systems, after which the development was taken over by the CiA. CAL is now generally accepted in the CAN world as the protocol for new developments. If it had been developed a few years earlier, most of the other protocols like SeleCAN, SDS and DeviceNet would not exist now.

ANOTHER STEP FURTHER

Despite the acceptation of CAL, it quickly became clear that several important elements are still missing. The reason is that CAL only specifies the network messages, but no application functionality (i.e. control of I/O). This makes it impossible to use equipment from various suppliers without a large amount of programming work in the application. Therefore, no standard firmware can be developed that can control CAL modules.

In the real-time market, this is less of a problem, because here an application must always be manually programmed. This is a problem for PLC suppliers because it is more the habit in this market to supply "plug-and-play" controllers. And this is only possible if there is far-reaching standardization of the functionality of all modules, which is the case for some types of application areas. The document that describes such an application area is called a "profile". A number of companies (members of the CiA) have taken the initiative to develop a number of "CAN/Open" profiles, for example for remote I/O, motion, and MMI's (man-machine interfaces).

The development of CAN/Open is actually the next logical step in the development of CAN: first the chips, then CAL, and now the application profile. Although initially several years behind, CAN has now finally caught up with Profibus and Interbus/S at protocol level. In the course of time, various profiles have also been designed on other fieldbuses, such as Interbus/S and DeviceNet. What has now been developed in the framework of CAN/Open strongly resembles what was made for Interbus/S in the past, which in turn was influenced by Profibus.

FUNCTIONS OF CAN/OPEN

CAN/Open is actually not a protocol in itself, but merely a description of how CAL must be used for a certain type of functionality. There is a two-stage description of CAN/Open: a base document (DS301) describes the general structure of a piece of equipment, and additional documents (DS40x) describe a certain type of functionality (e.g. for remote I/O, motion, MMI, etc.). These documents together describe how CAL must be used to implement that type of functionality. Here, one must think of matters such as:

- Configuration;
- Initialization;
- Run-time behavior;
- Error handling.

Why is this required? CAL itself only determines which services of the protocol can be used for this, but not in which sequence. One example of this is the start-up behavior of a module (figure 2). CAN/Open also determines which bit means what in data structures. Furthermore, a structure is determined in which the characteristics of a module can be read (over the network). A controller module can then cope with the variability in functionality between modules and between equipment from different suppliers. This structure is called "Object Dictionary".

As the same says, an object dictionary (OD) is a description of all objects that are present in a piece of equipment. The structure of the OD is as follows:

- General part: the same structure for all profiles;
- Technical part: specific part for the application area;
- Free part: intended for supplier-specific functions.

The general part is necessary because it makes possible the simultaneous use of equipment with different profiles on the same network. Companies from the rel-

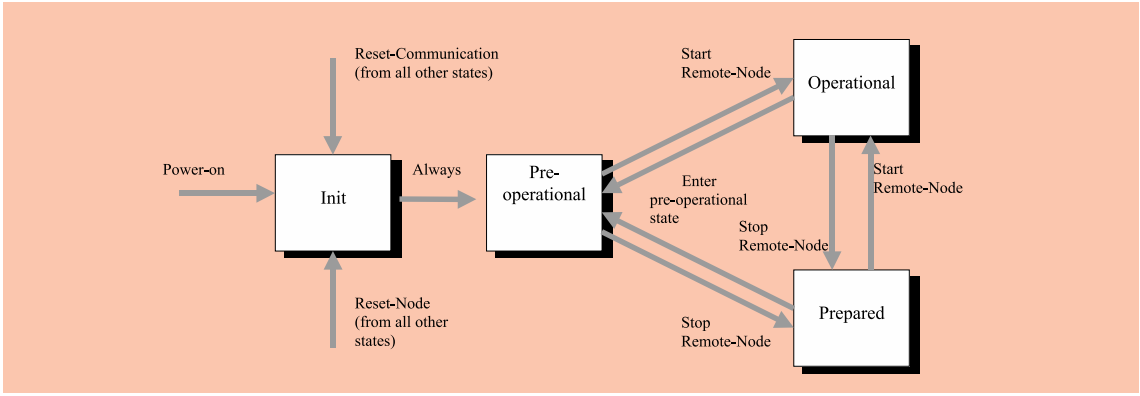

Figure 2: the start-up behavior of a module

evant branch mostly fill in the technical part of a profile. These companies together determine the profile in a working group of the CiA. For example, for a motor control this working group can determine how a number of parameters are interrelated (figure 3). In this way, there can be no differences in the control of equipment from different suppliers.

Finally, each supplier has the freedom to incorporate specific features. This means that there must always be a possibility for an application programmer to address these capabilities. It is precisely here where suppliers can still distinguish themselves, and this is therefore often a reason for a user to buy exactly that piece of equipment.

DETERMINING STANDARDS

CAN/Open is a standard of the CiA, and therefore has no official legal national or international status. There are, however, no comparable national or international standardisation activities. Therefore, the CiA standards are a "defacto" standard in the CAN world, and are also well known. In the past years, several CiA standards have been laid down (table 1).

All documents are still "Draft Standard", despite the fact that some have been in existence for years. They can be ordered from the CiA (address is given at the end of this article). DS150 is a new development, with a somewhat confusing name because it has no relation to the OSI session layer (5). The document describes the behavior of a piece of equipment that can be switched to a "low power mode" between messages, and back to active after receiving a certain message. At the moment, there are a number of working groups for CAN/Open profiles. The corresponding documents covering these profiles have been given numbers in the 400 series (table 2).

The motion profile is strongly oriented on what is now possible in Drivecom (which is a profile of Interbus/S). Considering the (growing) size, some motion experts regard it as overkill. Finally, there is another working group in the CiA for hydraulics (mainly Swedish companies) who, however, do not operate under the CAN/Open flag. Dutch companies are unfortunately hardly active in the CiA.

EVOLUTION OF CAN/OPEN

Will it stop here? Probably not. On other fieldbuses, where the standardization activities (such as Interbus/S) have evolved further, there are profiles for

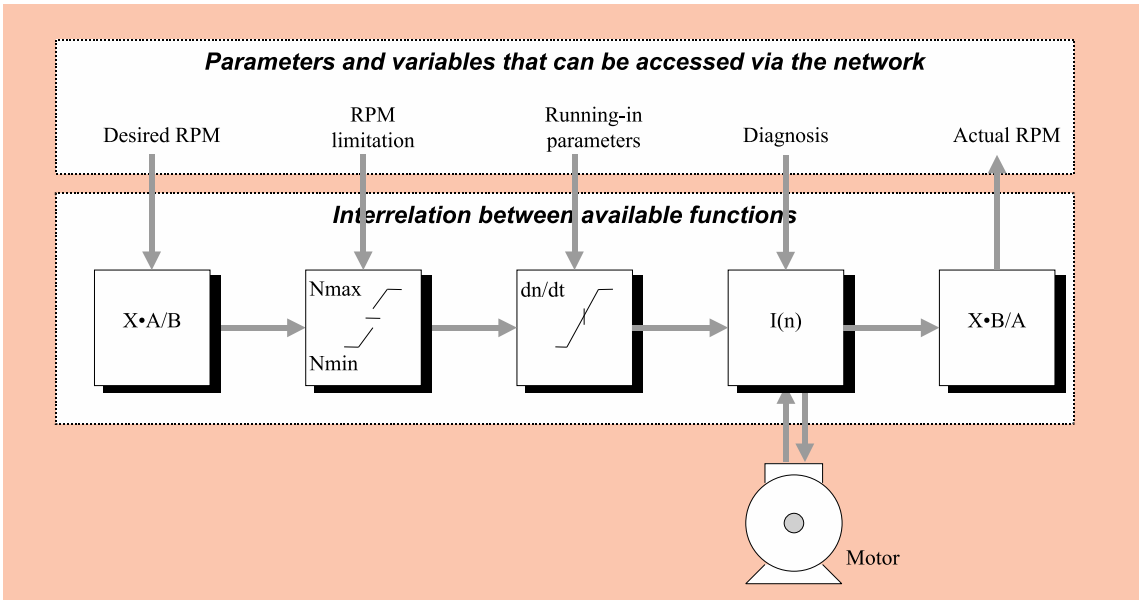

Figure 3. For a motor control this working group can determine how a number of parameters are interrelated

MMIs, robots, welding equipment, encoders, etc. It is expected that even more profiles will be added to CAN/Open in the coming years. However, the more complex the functionality to be standardized, the more difficult it becomes to design a profile. The experiences with other fieldbuses (including Profibus) show that if the profiles are too complex, they are not actually implemented by anybody. But if the profile is too simple, than many aspects outside the profile must be programmed differently for each supplier, so that we get a certain uncontrolled growth of possibilities. And the purpose of a profile is precisely to prevent this!

To what extent the CAN/Open standard is "ready" is still to be seen. The experiences that have been gained with CAN/Open are still being processed in newer versions of the standard. At the end of 1996 the CiA will release a more definitive version of the standard, and it will be frozen for several years.

COMPETITORS OF CAN/OPEN

In the CAN world, there are various protocols that provide more or less the same functionality as the CAN/Open profile for remote I/O. This concerns systems that do roughly the same: access analogue and digital I/O. We therefore see the same functions returning every time.

Competing protocols of CAN/Open are mainly DeviceNet, SDS and SeleCAN. Despite the fact that all three protocols are based on CAN, it is not possible to simultaneously run them on the same network. Each protocol assumes that no other protocols are being run. The most important limitation here is the allocation of CAN identifiers. It might only be possible to run something in parallel on a very incidental basis, whereby the structure of the network is known beforehand. It is also possible to control remote I/O without a protocol when the SLIO-controllers (Serial Link I/O) of Philips Semiconductors are used. These controllers have their own analogue and digital I/O integrated on the chip, but a disadvantage is that only 16 can be simultaneously used on a network.

One important difference between CAN/Open on the one hand and DeviceNet and SDS on the other is that these are complete systems, while CAN/Open is no more than a protocol. By "systems" we mean that Allen-Bradley and Honeywell supply not only the protocol, but also the I/O, the master module, cables, diag-

Standard	*Contents*	*Status*	*Size*	*Costs*
DS102	Physical layer specification	Rev 2.0	4 pages	DM 10
DS150	Session layer specification	Rev 1.0 is being designed.	Still being designed.	Not yet available
DS201..207	CAL specification	Rev 1.1 dd. 1/2/96	150 pages	DM 120
DS301	CAN/Open communication profile	Rev 2.0 dd. 22/9/95	62 pages	DM 100

Table 1. CiA Standards

nostic equipment and manuals, etc. CAN/Open has not yet reached this status, but this will not take long. By issuing a special catalogue of CAN/Open equipment, the CiA indicates the importance of recognizing a systems approach for the users.

Moreover, CAN/Open still has competitors in the form of other fieldbuses, such as AS-Interface, Profi-bus/DP and Interbus/S. From the user's viewpoint, we then get the question of: what does CAN/Open offer that all those other systems do not offer? The other systems mentioned are typical master/slave systems, whereby a (central) master handles the I/O on the slaves. CAN/Open, however, offers extra possibilities because CAN is a multi-master system. It is therefore possible that modules can communicate directly with each other. This makes control of a distributed system much easier (and faster). CAN/Open has these possibilities because it uses CAL. Here, CAN/Open offers characteristics that were mainly available in LON up to now. However, LON has the advantage that it is further evolved than CAN/Open (at this moment).

LOCATION OF CAN/OPEN IN THE PROTOCOL STACK

Given the existence of CAN/Open with relevant profiles and all other CAN protocols, the question arises: how do all these protocols, profiles, norms and standards interrelate? Figure 4 gives an overview of this. The arrows indicate the interface(s) to a user's application. For example, some applications require direct access to the CAN-bus. Others use CAL or go indirectly via CAN/Open. Profile libraries can be placed above this.

It is clear that CAN/Open with CAL and the appropriate profiles offers the same functionality as DeviceNet, SDS and SeleCAN. This must be carefully taken into account with the purchase of equipment, because these protocols cannot be used together on the same network. Also, in industry, CAN 2.0A is always used with ISO-11898; version 2.0B has a longer identifier (29 bits instead of 11) and is mainly used in the automotive industry.

From figure 4, it appears that the use of CAN is certainly no sinecure for a random (new) user. It is expected that this situation will continue for some time yet. It is not expected that DeviceNet and SDS will be withdrawn in favor of CAN/Open; these systems are already far too advanced for this. Moreover, the market for DeviceNet and SDS is mainly in the USA, while German companies mainly promote CAN/Open (at the moment).

SeleCAN appears to have lost the fight, because the inventor (Selectron) has gone over to CAN/Open himself. The other protocols (e.g. LBS and MiCAN) are strongly related to a specific branch of industry, and are therefore only interesting for specific applications

Standard	*Profile contents*	*Status*	*Size*	*Costs*
DS401	I/O	Rev. 1.3 dated 1/3/96 End of '96 definitive	114 pages now, extension expected	DM 100 DS402
Motion	First version at Hannover Messe 96	400 pages	n.a. DS403	Human/machine
interface (HMI)	Working group started 9/96	n.a.	n.a. DS404	Regulators
Working group started 9/96	n.a.	n.a. DS405	PLCs + intelligent controllers	Working group started
9/96	n.a.	n.a. -	Printing presses	Working group outside
the CiA	n.a.	n.a.		

Table 2. Standardization Documents CiA

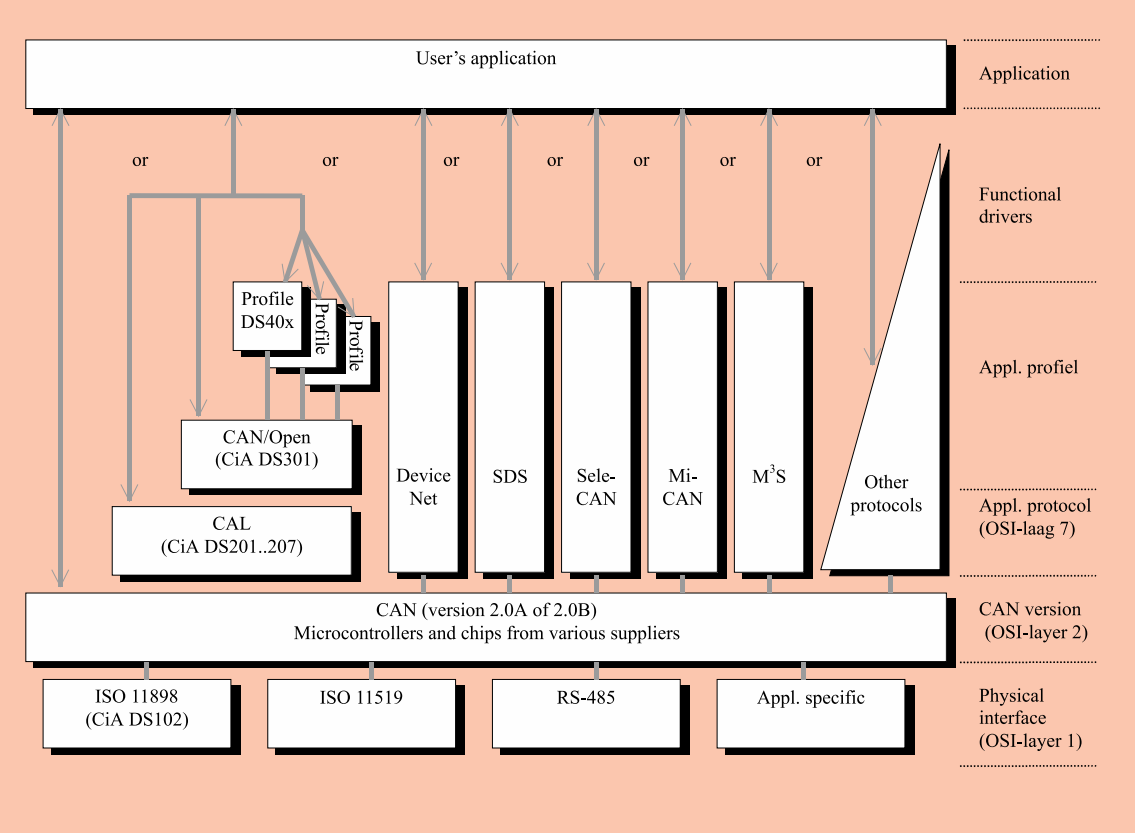

Figure 4. The use of CAN is certainly no sinecure for a random

and companies (agricultural for LBS, and maritime for MiCAN).

SOFTWARE THAT NEEDS TO BE MADE

In principle, no software is needed for the support of a CAN/Open profile. A profile only describes how existing CAL services must be called and in which sequence, and which bits must be filled in to program a certain function. Therefore, a user who is armed with a CAL implementation can program everything himself. This is quite possible for simple networks.

In practice, however, this does not in this way. Many technical matters with regard to profiles must always be done in the same way, for example the start-up of a module (figure 2). It will take quite a lot of work to program this anew each time. We therefore often get libraries that take care of these tasks.

It is also possible that an application uses equipment from different suppliers, who may implement different profiles; for example, a machine with I/O, an MMI, and a motor control. It must therefore be possible that the application program can simultaneously use several of these profile-libraries.

THE NEAR FUTURE

CAN/Open is still a new branch in the forest of CAN software. It is, however, a necessity: without this step, CAN will never be a serious competitor of other systems that have been in existence for longer, such as Profibus and Interbus/S. Moreover, CAN/Open offers extra functionality that is not available in other systems, thanks to the specific characteristics of CAN that are still available for the user. ■

Rob Hulsebos joined Philips in 1986. After his study Computer Science and specializing in data-communications, he developed several network modules for the P8 PLC family, such as PPCCOM, Profibus, CAN, Modbus, TCP/IP, Bitbus and AS-Interface. His experiences with CAN and its protocols are the base for this article.