December 2014 B 25361

CAN Newsletter

Hardware + Software + Tools + Engineering

Applications

20th anniversary: CANopen in the hands of CiA

Detecting and counting unwanted particles

CAN FD: Improved residual error-rate

www.can-newsletter.org

www.esd.eu

CAN-PCIe/402

CAN-USB/400

CAN-PCI/400

Ethernet

CAN-USB/2

Gateways

All you CAN plug

CAN-PCIe/402

- up to 4 high performance PCI Express CAN interfaces
- DMA busmaster
- Powered by esd Advanced CAN Core (esd-ACC)
- MSI (Message Signaled Interrupt) support
- Electrically isolated
- Provides high resolution hardware timestamp

CAN-USB/400

- 2 high performance CAN-USB interfaces
- Powered by esd Advanced CAN Core (esd-ACC)
- USB 2.0 with high speed data rates of 480 Mbit/s
- Electrically isolated
- Provides high resolution hardware timestamp
- Error injection for advanced diagnostic
- IRIB B timecode as option

CAN-PCI/400

- up to 4 high performance CAN interfaces
- Powered by esd Advanced CAN Core (esd-ACC)
- Electrically isolated
- Provides high resolution hardware timestamp
- Error injection for advanced diagnostic

CAN-PCI104/200

- PCI104-CAN interface
- One or two CAN interfaces for PCI104 bus

EtherCAN

- 10/100 BaseT ETHERNET-CAN Gateway
- Electrically isolated
- Configuration and Diagnostics by webbrowser

CAN-USB/2

- CAN-USB interface
- Intelligent CAN interface with ARM 7
- USB 2.0 with high speed data rates of 480 Mbit/s
- Electrically isolated
- Provides high resolution hardware timestamp

Gateways

- EtherCAT-CAN
- PROFINET-CAN
- PROFIBUS-CAN
- PROFIBUS-DeviceNet
- EtherNet/S7-CAN

Operating Systems

esd supports the real-time multitasking operating systems VxWorks, QNX, RTX, RTOS-32 and others as well as Linux and Windows 32/64Bit systems

CAN Tools

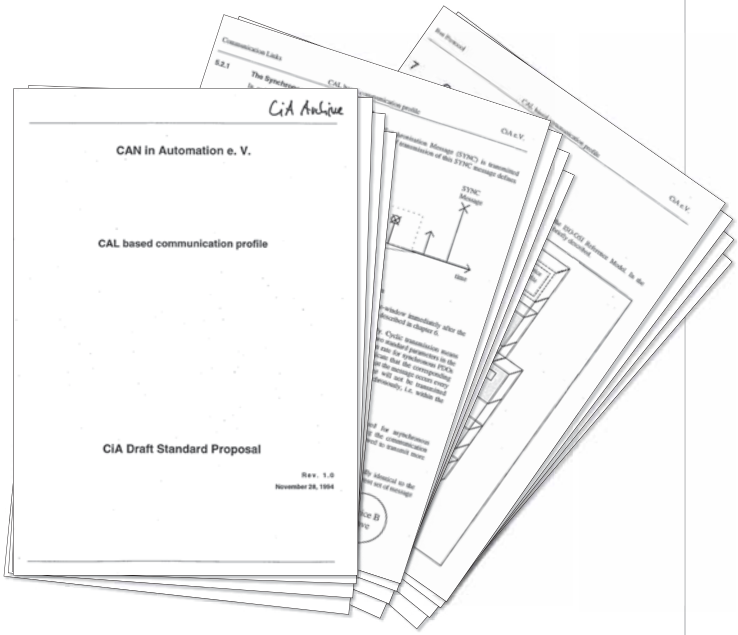
- CANreal: Display and recording of CAN message frames
- CANplot: Display of online/offline CAN data
- CANrepro: Replay of pre-recorded CAN message frames
- CANscript: Python scripting tool to handle CAN messages
- COBview: Analysis and diagnostics of CANopen nodes

The tools are free of charge on the driver CD or can be downloaded at www.esd.eu

20th anniversary: CANopen in the hands of CiA

In November 1994, CiA published the very first version of the CANopen specification: CiA 301 was one of the most successful Esprit research projects. After all this time, CANopen is still unique in many ways.

In the beginning, the CANopen specification was named CAL-based communication profile for industrial systems. It was developed under the umbrella of Esprit (European Strategic Program on Research in Information Technology), a research program of the European Community. The title of the project 7302 was ASPIC, short for "Automation and Control Systems for Production Units using an Installation Bus Concept". The objective was to develop control architectures and devices to enable flexible and modular combination of existing manufacturing cell units. The researchers led by Dr. Mohammad Farsi (University of New Castle) and Stefan Reitmeier (Bosch) decided to use the CAN Application Layer (CAL) protocol, developed by CiA. CAL was a pure application layer approach according to the OSI (open systems interconnection) model. However, it was in some respect an academic approach and had various fathers: Main contributions came from Tom Suters (Philips Medical Systems), as well as Prof. Dr. Konrad Etschberger and Prof. Dr. Wolfhard Lawrenz, both working at German universities for applied science. Of course, other CiA members had also contributed ideas.



CiA Archive

CAN in Automation e. V.

CAL based communication profile

CiA Draft Standard Proposal

Rev. 1.0
November 28, 1994

Figure 1: The first edition of the CiA 301 specification contained only 60 pages, but it was not as complete as it is nowadays

The ASPIC project's objective was to develop an application layer that was easy to implement, dedicated to embedded machine control. Under the leadership of Bosch, several companies (Moog, ADL Automation, and J.L. Automation) and institutes (Newcastle university and Reutlingen university of applied science) specified the first version of what is today known as CANopen. Main contributors were Dr. Mohamad Farsi and Prof. Dr. Gerhard Gruhler. The first version already defined PDOs (process data objects) and SDOs (service data objects). The synchronous transmission of PDOs was introduced as well as Network Management (NMT) and Emergency messages.

In the early days of CANopen, CAN Remote Frames were still in favor, which is why Node Guarding was based on them. Later, Node Guarding was substituted by the Heartbeat mechanism. The first CANopen networks also used remotely requested PDOs. Nowadays, CiA recommends not using remote frames at all.

▷

> **"In the beginning, not all CiA members were in favor of CANopen. Many preferred non-standardized application layers, so called layer-2 protocols.**

Where have the years gone?

It seems to me like it was yesterday that the first CANopen documents came to my desk for editing. Now, 20 years later, CiA's secretaries have to maintain more than 15000 pages of CANopen specifications. The success of CANopen has many fathers and a few mothers. It was a joint success, mainly of small and medium-sized companies and some big machine building enterprises. It constitutes one of the rare cases of a company-independent communication standard – not driven by marketing money but by a community of individuals. And even after 20 years, the story has not come to an end. More development will be necessary in the coming years. I am sure that CAN FD will have an impact on existing profile specifications and will initiate further applications, which will benefit from the larger payload and increased throughput.

Holger Zeltwanger

The CANopen specification published as CiA 301 was one of the most successful Esprit research projects. One of the reasons was that the specification was handed over to CiA for further developments and maintenance. From the beginning, several companies implemented the higher-layer protocol in real applications. Of course, several reviews and updates were necessary before CANopen became a stable specification. Version 3.0 was the first release used in products and systems. This version was valid from 1996 to 1999. Some applications still use this version today.

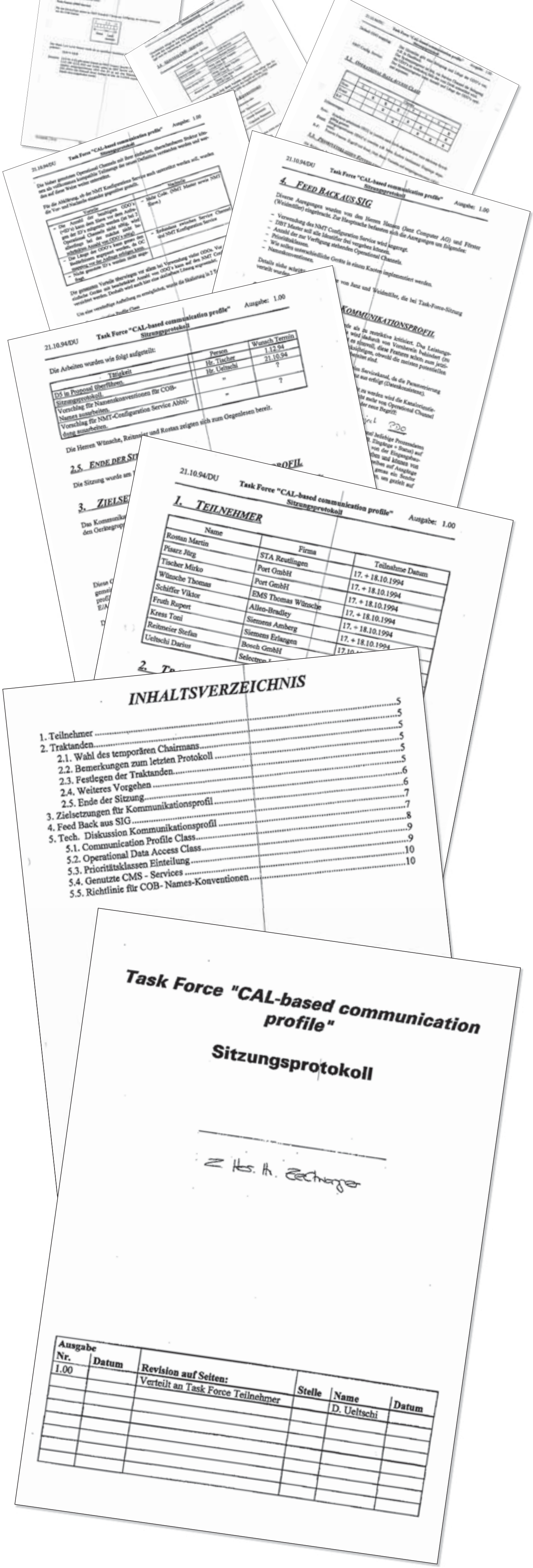
> “ *The CANopen success story is unique, because it was not promoted by one big supplier.*

CANopen can be regarded as the application layer of small and medium-sized suppliers. It is the only independent industrial communication system not promoted by one market-leading company. It can also be regarded as the solution of system designers, because some machine builders have chosen this approach to be independent from the suppliers. Among these machine builders are Heidelberger and Siemens Healthcare. In 1995, CiA presented the very first CANopen multi-vendor demonstrator equipped with products from Moog, Selectron and others at its Hanover fair booth.

Holger Zeltwanger

20th anniversary of Devicenet

Devicenet is also 20 years old. Originally it was developed by Allen-Bradley. As early as 1992, Allen-Bradley and Honeywell, together with the Cincinnati Milacron machine builder, started specifying a CAN-based network solution. In March 1994, Allen-Bradley introduced Devicenet at the ICEE show in Chicago. One year later, the company initiated the Open Devicenet Vendors Association (ODVA). At that time, CiA also promoted Devicenet and sold the specification in Europe. But this cooperation ended after a few years.



21.10.94/DU Task Force "CAL-based communication profile" Sitzungsprotokoll Ausgabe: 1.00

Die Arbeiten wurden wie folgt aufgeteilt:

Tätigkeit	Person	Wunsch Termin
D5 in Proposal überführen.	Hr. Tischer	1.12.94
Sitzungsprotokoll.	Hr. Ueltschi	21.10.94
Vorschlag für Namenskonventionen für COB-Namen ausarbeiten.	"	?
Vorschlag für NMT-Configuration Service Abbildung ausarbeiten.	"	?

Die Herren Wünsche, Reitmeier und Rostan zeigten sich zum Gegenlesen bereit.

2.5. Ende der Si…

Die Sitzung wurde am …

3. Zielse…

4. Feed Back aus SIG

21.10.94/DU Task Force "CAL-based communication profile" Sitzungsprotokoll Ausgabe: 1.00

1. Teilnehmer

Name	Firma	Teilnahme Datum
Rostan Martin	STA Reutlingen	17. + 18.10.1994
Pisarz Jürg	Port GmbH	17. + 18.10.1994
Tischer Mirko	Port GmbH	17. + 18.10.1994
Wünsche Thomas	EMS Thomas Wünsche	17. + 18.10.1994
Schiffer Viktor	Allen-Bradley	17. + 18.10.1994
Fruth Rupert	Siemens Amberg	17. + 18.10.1994
Kress Toni	Siemens Erlangen	17. + 18.10.1994
Reitmeier Stefan	Bosch GmbH	17.10.…
Ueltschi Darius	Selectron …	

INHALTSVERZEICHNIS

1. Teilnehmer 5
2. Traktanden 5
   2.1. Wahl des temporären Chairmans 5
   2.2. Bemerkungen zum letzten Protokoll 5
   2.3. Festlegen der Traktanden 5
   2.4. Weiteres Vorgehen 5
   2.5. Ende der Sitzung 6
3. Zielsetzungen für Kommunikationsprofil 6
4. Feed Back aus SIG 7
5. Tech. Diskussion Kommunikationsprofil 7
   5.1. Communication Profile Class 8
   5.2. Operational Data Access Class 9
   5.3. Prioritätsklassen Einteilung 9
   5.4. Genutzte CMS - Services 10
   5.5. Richtlinie für COB- Names-Konventionen 10

Task Force "CAL-based communication profile"

Sitzungsprotokoll

Z. Hd. H. Zeltwanger

Ausgabe Nr.	Datum	Revision auf Seiten:	Stelle	Name	Datum
1.00		Verteilt an Task Force Teilnehmer		D. Ueltschi	

Figure 2: The first meeting minutes of the CiA Task Force “CAL-based communication profile” were written in German; today all CiA documents are written in English

Your Source For

System Integration

- Colour and monochrome displays
- Freely configurable HMI graphics, state machines, diagnostics, CAN messaging and scripting/activity programming
- Compatible with CAN 2.0B, SAE J1939, CANopen, freeform
- Ultra-bright, bonded & coated LCD:
  - Maximum visibility, works with polarized sunglasses
- Data logging and graphical trending
- Reduce wire harness cost with IX3212 Power Distribution Modules:
  - 12 outputs: 15A per output/70A per module, on-off/PWM/H-bridge
  - 12 digital & 8 analogue inputs.
- Robust and reliable: suitable for use outside the cab
  - Class-leading environmental protection: IP67, -40 to +85˚C

Murphy's PowerView displays integrate your control systems and machine data in a powerful, rugged package. With superb environmental specs, bonded screen and sealed connectors, it provides durable performance even in the harshest applications.

Seamlessly add CAN-controlled solid-state I/O and power distribution where you need it with the PowerCore IX3212. This Power Distribution Module directly drives high-power motors, lamps, actuators and loads via a four-wire power and CAN connection. Improve your control, reliability and load diagnostics while saving on wiring and labour costs.

PowerView™
Freely Configurable Displays

PowerCore™
Controller, Input/Output &
Power Distribution Modules

Integrate your total package with Murphy products!

MURPHY®
by ENOVATION CONTROLS

ENOVATION CONTROLS CORPORATE HEADQUARTERS
5311 S 122ND EAST AVENUE
TULSA, OK 74146

ENOVATION CONTROLS – SAN ANTONIO OFFICE
5757 FARINON DRIVE
SAN ANTONIO, TX 78249

ENOVATION CONTROLS – HOUSTON OFFICE
105 RANDON DYER RD
ROSENBERG, TX 77471

ENOVATION CONTROLS LTD. – UNITED KINGDOM
CHURCH ROAD LAVERSTOCK
SALISBURY SP1 1QZ UK

MURPHY ECONTROLS TECHNOLOGIES (HANGZHOU) CO, LTD.
77 23RD STREET
HANGZHOU ECONOMIC & TECHNOLOGICAL DEVELOPMENT AREA
HANGZHOU, ZHEJIANG 310018 CHINA

DOMESTIC SALES & SUPPORT

ECONTROLS PRODUCTS
PHONE: 210 495 9772
FAX: 210 495 9791
EMAIL: INFO@ECONTROLS.COM
WWW.ECONTROLS.COM

MURPHY PRODUCTS
PHONE: 918 317 4100
FAX: 918 317 4266
EMAIL: SALES@FWMURPHY.COM
WWW.FWMURPHY.COM

MURPHY CONTROL SYSTEMS & SERVICES
PHONE: 281 633 4500
FAX: 281 633 4588
EMAIL: CSS-SOLUTIONS@FWMURPHY.COM

MURPHY INDUSTRIAL PANEL DIVISION
PHONE: 918 317 4100
FAX: 918 317 4124
EMAIL: IPDSALES@FWMURPHY.COM

INTERNATIONAL SALES & SUPPORT

UNITED KINGDOM
PHONE: +44 1722 410055
FAX: +44 1722 410088
EMAIL: SALES@ENOVATIONCONTROLS.EU
WWW.FWMURPHY.EU

CHINA
PHONE: +86 21 6237 5885
FAX: +86 21 6237 5887
EMAIL: APSALES@FWMURPHY.COM

LATIN AMERICA & CARIBBEAN
PHONE: 918 317 2500
EMAIL: LASALES@FWMURPHY.COM

SOUTH KOREA
PHONE: +82 70 7951 4100
EMAIL: SKOREASALES@FWMURPHY.COM

INDIA
PHONE: +91 91581 37633
EMAIL: INDIASALES@FWMURPHY.COM

ISO 9001
REGISTERED

bsi.

USA (TULSA) - ISO 9001:2008 FM 28221
USA (ROSENBERG) - ISO 9001:2008 FM 28221
UK-ISO 9001:2008 FM 29422
CHINA-ISO 9001:2008 FM 523851
Registered Facilities

Printed in the USA

20th anniversary: CANopen in the hands of CiA

Imprint

Publisher
CAN in Automation GmbH
Kontumazgarten 3
DE-90429 Nuremberg

publications@can-cia.org
www.can-cia.org
www.can-newsletter.org

Tel.: +49-911-928819-0
Fax: +49-911-928819-79

CEO Holger Zeltwanger
AG Nürnberg 24338

Printed in Germany
Hard copies: 3000
Soft copies: 3000

Editors
(pr@can-cia.org)

Annegret Emerich

Cindy Weissmueller

Holger Zeltwanger
(responsible according
to the press law)

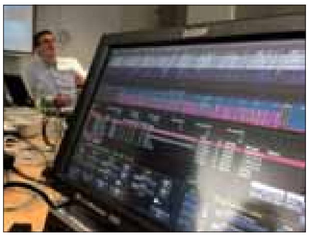
Layout
Nickel Plankermann

Advertising manager
Gisela Scheib
(responsible according
to the press law)

Distribution manager
Julia Adolf

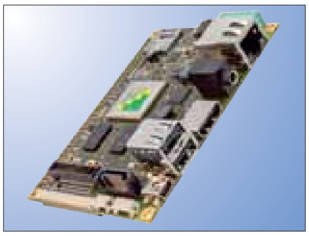
© Copyright
CAN in Automation GmbH

Applications



CAN FD

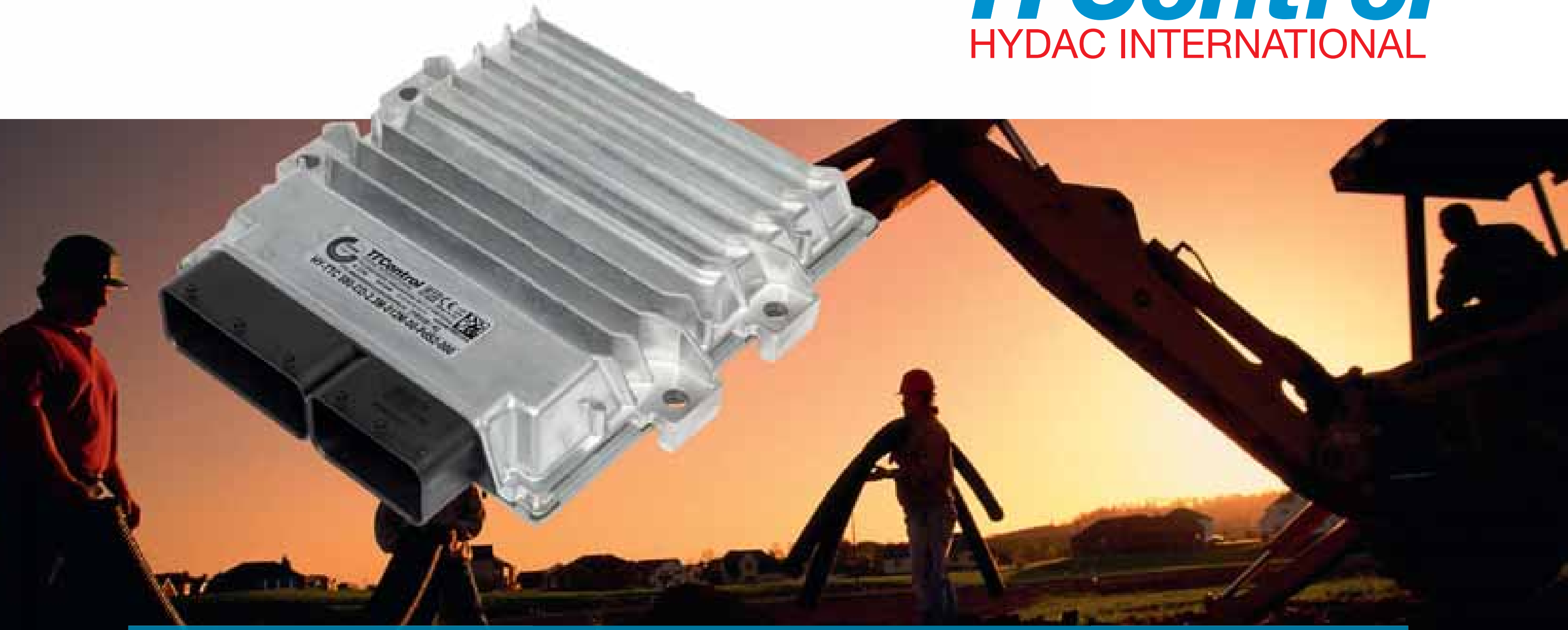
Devices



Software

Tools

TTControl
HYDAC INTERNATIONAL



Powerful Control Units for High-End Applications: HY-TTC 500 Family

Flexibility & Usability

- Single controller for whole vehicle for centralized architectures
- Extensive I/O set with multiple software configuration options per pin
- Open programming environments C, CODESYS® V3.x and CODESYS® V3.x Safety SIL 2

Safety

- TÜV-certified according to IEC 61508 (SIL 2) and EN ISO 13849 (PL d)
- ISO 25119 AgPL d certifiable
- CODESYS® Safety SIL 2 including support for CANopen® Safety Master and separation of safe / non-safe code
- Safety mechanisms in hardware to minimize CPU load
- Up to 3 output groups for selective shut-off in case of safety relevant fault

Connectivity

- Up to 7 CAN interfaces
- Automatic baudrate detection and configurable termination for CAN
- Ethernet for download and debugging purpose

Performance

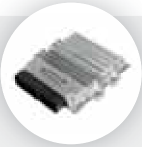
- 32 bit / 180 MHz TI TMS570 dual core lockstep processor (ARM architecture)
- Safety companion

Robustness

- Automotive style housing suited for rough operating conditions
- Total current up to 60 A

bauma China 2014
Meet us at bauma China
November 25-28, 2014

www.ttcontrol.com/HY-TTC-500-Family



Safety
Certified
ECUs

General
Purpose
ECUs

I/O Slave
Modules

Safe I/O
Modules

Operator
Interfaces

Up to six CAN FD cores on one micro-controller

Starting with the Aurix family, Infineon offers CAN FD for all its micro-controllers. The CAN FD IP supports up to 64 data bytes and mixing of classical CAN messages and CAN FD messages.

Author
Ursula Kelling

Infineon Microcontrollers
Im Campeon 1-8
DE-85579 Neubiberg
Tel.: +49-89-234-83287
Fax: +49-89-234-955-6811

Link
www.infineon.com

Starting with the Aurix family, Infineon offers CAN FD for all devices. The CAN FD IP supports up to 64 data bytes and mixing of classical CAN messages and CAN FD messages.

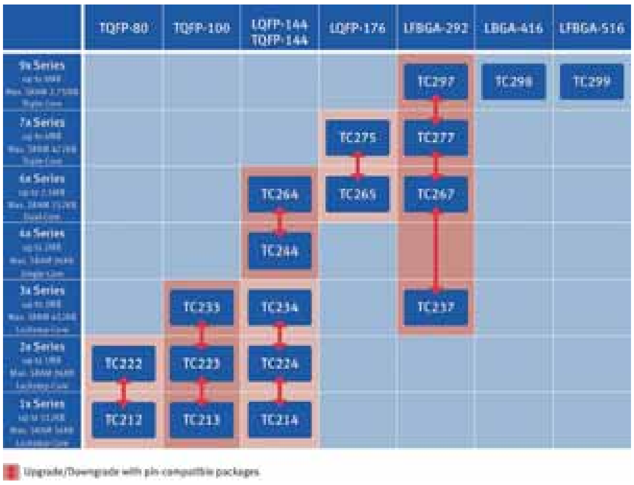
Depending on the device, up to 6 nodes support CAN FD. Typical application use cases like 500 kbit/s arbitration speed and 2 Mbit/s data speed can be realized. The data segment can be used up to 5 Mbit/s. CAN FD frames can include up to 64 data bytes.

With Aurix, MultiCAN+ has been introduced, which is a further development of the MultiCAN module. The module has always supported features like automatic rerouting of messages (gateway mode) and flexible Fifo structures. All devices come with an asynchronous clock input for the bit-rate clocking, enabling the nodes to be driven by either the system clock, directly from an oscillator, or by the precise ERAYPLL configured to 80 MHz. Each message object can take part in a receive time-out. The receive time-out counter exists once per node. This opens the possibility to react if a specific message no longer arrives. In automotive applications this message is part of the network management messages. To trigger messages in equidistant time distances, three messages per node can be configured to be transmitted automatically. For example in case of an operating system alarm, the contents of the message objects can be updated. After the execution of the interrupt, the CPU gets the IDLE instruction and goes back in IDLE mode. By using this feature, the transmit request is set by the module at the right point in time. As an interrupt can be triggered on a received message, these functions can be used to support pretended networking.

Figure 1: Aurix can mix classic CAN and CAN FD messages

CAN FD integration into CAN nodes

The integration of the CAN FD protocol is quite straightforward: After enabling the module, the device remains in classical CAN mode. CAN FD can be enabled for every single node. Only in case CAN FD is enabled for the node, the registers additionally needed for CAN FD become active and can be programmed. Once enabled, bit timing can be configured for arbitration and data phase separately. The transmitter delay compensation is configurable automatically or manually, dependent on the setting. ▷



	TQFP-80	TQFP-100	LQFP-144 TQFP-144	LQFP-176	LFBGA-292	LBGA-416	LFBGA-516
9x Series					TC297	TC298	TC299
7x Series				TC275	TC277		
6x Series			TC264	TC265	TC267		
4x Series			TC244				
3x Series		TC233	TC234		TC237		
2x Series	TC222	TC223	TC224				
1x Series	TC212	TC213	TC214				

🟥 Upgrade/Downgrade with pin-compatible packages

Figure 2: Overview over the Aurix family

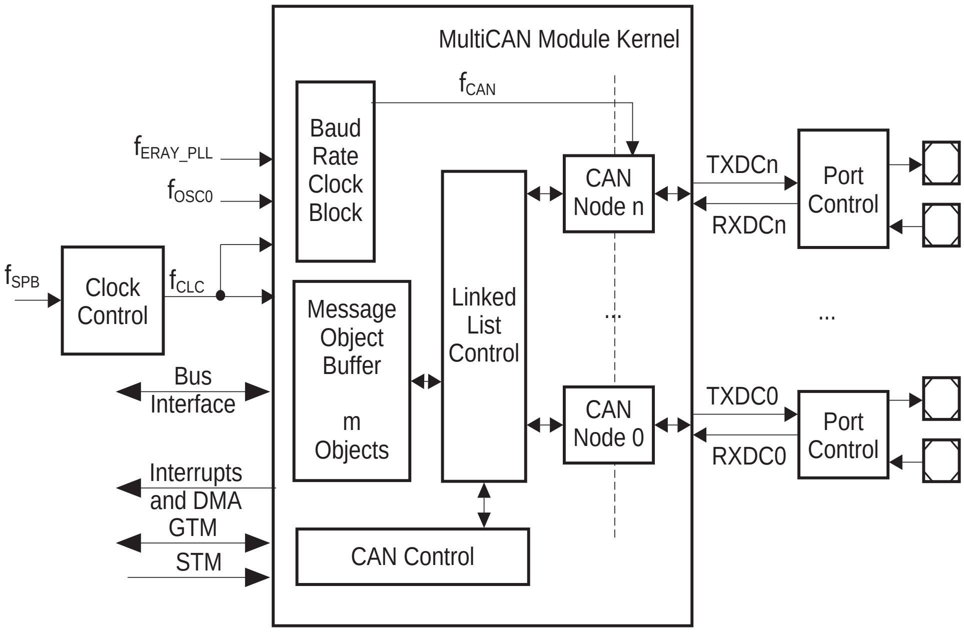


Figure 3: Structure of the MultiCAN+ module

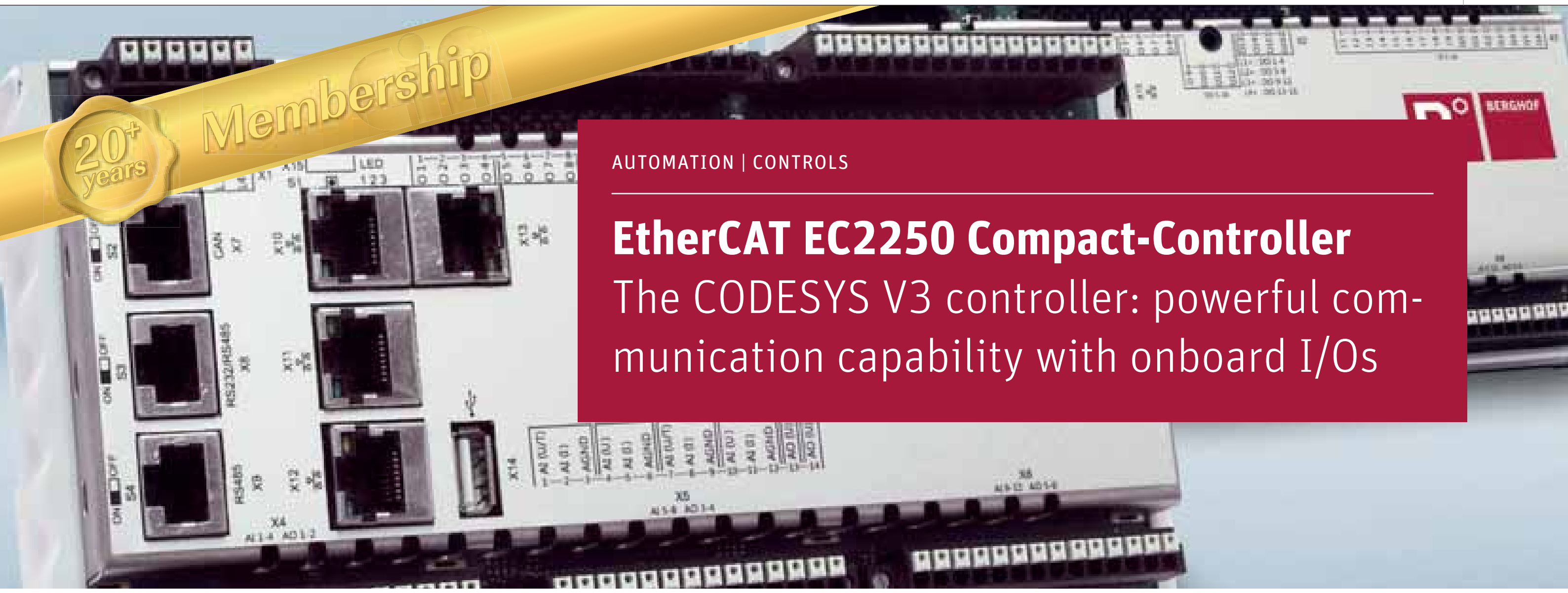
CAN FD integration into the message structure

The Linked List Structure enables the integration of 64 bytes. If the additional bytes are enabled for a message object, the message object itself points to the additional message objects used as data space. A concatenation of three message objects gives the 64 bytes needed for example for a flashing application.

In every message object chosen for transmission, the message can be configured to be sent in classical CAN mode or in CAN FD mode. For example ISO 11898-6 compliant devices, so called partial networking transceivers, still need the classical CAN mode, whereas the rest of the bus might run with CAN FD. Inside every message object the mode can also be configured, if bit-rate switching is used.

The same bits used for configuration in case of transmission are used as status bits for reception. This enables software to check if the message has been received in the right mode. If not, the application layer can react accordingly.

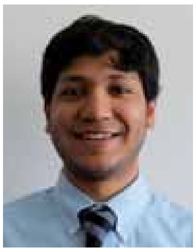
The MultiCAN+ implementation used in the Aurix family enables CAN FD with 64 data bytes. Mixing of classical CAN messages and CAN FD messages is supported. The module enables pretended and partial networking for automotive applications. ◀



Improving compatibility of Isobus devices

Isobus standardizes communication between tractors and implements. Still, Isobus communication does not always ensure compatibility. AEF has developed a system that will help prevent and resolve these issues.

Author



Juan Aguilar

Sontheim Industrie
Elektronik GmbH
Businees Development &
Application Engineer

Link
www.s-i-e.de

CAN Newsletter (print)
Automatic interoperability tests

For as long as agriculture has been around, people have been looking for ways to improve efficiency and yield by developing and using new technologies. This has led to the rise of many manufacturers of different types of agricultural equipment such as tractors, implements, displays, etc. As the number of manufacturers increased, farmers were able to purchase equipment to meet their specific needs. Unfortunately, this led to compatibility problems among different pieces of equipment, especially issues regarding communication between different devices. In an effort to alleviate these issues, equipment manufacturers along with organizations such as the International Organization for Standardization (ISO) have worked together to develop standard interfaces for different equipment: both physical and electrical. This led to the development of a standard for communication methods among different agricultural equipment parts called ISO 11783 – "Tractors and machinery for agriculture and forestry - Serial control and communications data network" – commonly known as the Isobus [1].

By adhering to the Isobus standard, equipment manufacturers strive to provide customers with equipment that will work properly with their equipment. However, this is not always the case. Sometimes a farmer will buy an implement or a device to install on their tractor from a different manufacturer and find that the implement's features do not work, and in some cases, the device does not work at all with the farmer's equipment. The farmer then contacts the manufacturer of the tractor or implement, but the technician has trouble finding the compatibility problem since the tractor or implement itself does not seem to have any issues. In this scenario, it is very difficult and time-consuming to find a solution to the problem. For this reason, the Agricultural Industry Electronics Foundation (AEF) has developed a system that will help prevent and resolve these compatibility issues and avoid this finger-pointing scenario. This system is the AEF Isobus Database and Isobus Check Tool system for agricultural equipment compatibility within Isobus.

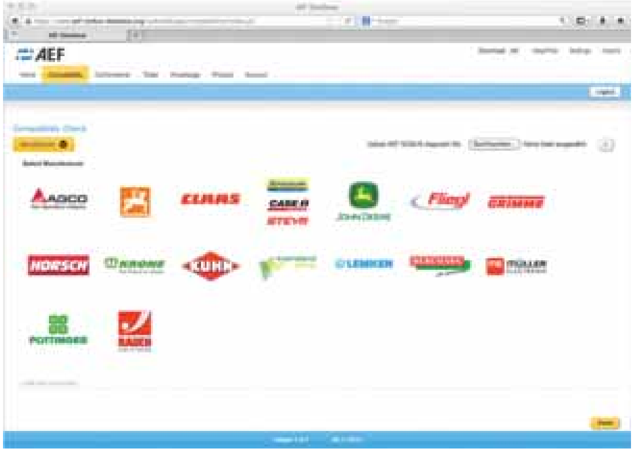


Figure 1: The Isobus Database shows various tractors, implements, and other agricultural equipment from manufacturers

AEF and Isobus

The Agricultural Industry Electronics Foundation (AEF) was established by a group of seven internation- ▷

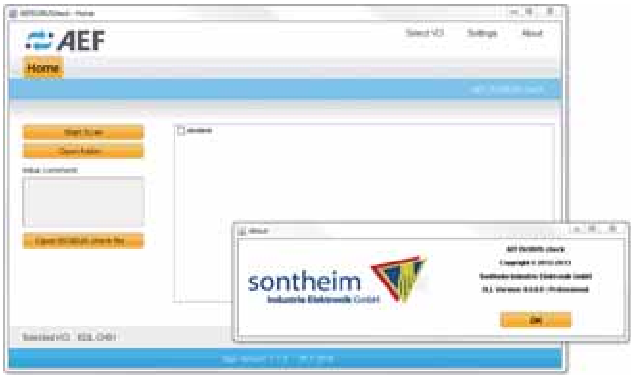


Figure 2: The Isobus Check Tool collects important diagnosis information of different ECUs on the bus

References
[1] ISO 11783: An electronic communications Protocol for agricultural Equipment
[2] AEF website (http://www.aef-online.org/en/)
[3] AEF database presentation (https://www.aef-isobus-database.org/isobusdb/docs/aef_presentation_en.pdf)

CAN Aerospace
ARINC
NMEA 2000
ISOBUS
CANopen
J1939

20+ years
Membership

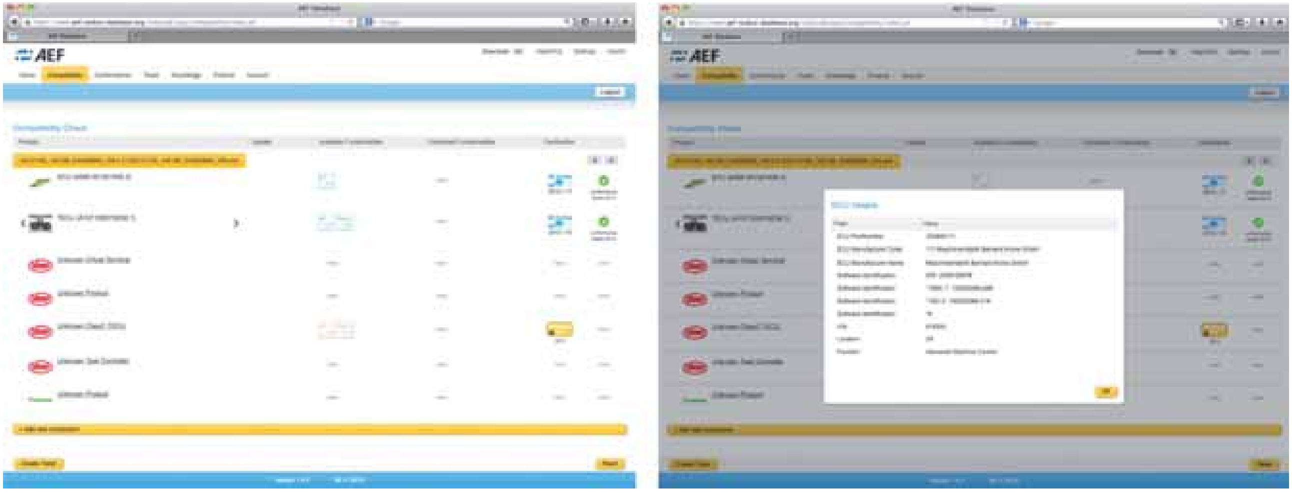


Figure 3: Scan data from the Isobus Check Tool uploaded to the AEF Database

al agricultural equipment manufacturers (Kverneland Group, Grimme, AGCO, John Deere, Pöttinger, Claas, and CNH) and two associations (VDMA, AEM) on October 28, 2008 as an independent international organization. Its aim is to provide resources and know-how for electronic systems in agriculture and to help with the adoption and execution of the Isobus standard. Since its inception, AEF has grown in membership to include more than 170 companies, associations, and organizations involved in electrical and electronic systems in agriculture and it has expanded its areas of interests to include Farm Management Information Systems (FMIS), electric drives, and camera systems [2].

The focus of Isobus is to standardize the communication between tractors and implements and to ensure full compatibility of data transfer between the different systems involved in farming. The use of such standardized interfaces and communication methods increases both efficiency and functionality of agricultural systems. The goal is to achieve plug-and-play functionality between different tractors, implements, and devices so that the farmer does not waste time, effort, and money searching for a compatible component to add to his system or trying to achieve full functionality from different components already in his system [1,2].

With the support of Isobus at the forefront of AEF's efforts, AEF aims to increase international acceptance and awareness of the standard, enhance customer benefits when using Isobus technology, and improve compatibility of Isobus products. The foundation also collects information about Isobus products for the service, marketing, and sales divisions of manufacturers and suppliers and promotes the acceptance of Isobus certified products worldwide [2]. It achieves these goals through different project groups and the development of tools that function as resources and support for manufacturers, suppliers, and farmers. Examples of such tools include the AEF Isobus Database and the AEF Isobus Check Tool. These tools help to mitigate compatibility issues among different tractor, implement, and device manufacturers as well as provide farmers with a valuable resource for selecting appropriate equipment.

Isobus Database

AEF has developed tools to help agricultural equipment users answer the following questions: Which implement/tractor should I purchase to take full advantage of the possible functions with my current system? and Are my current implements/tractors Isobus certified? [3] These are questions that farmers are faced with when evaluating their current system and when planning to add new equipment. For example, a person with a tractor from company X wants to buy an implement for their application. It is difficult for that person to truly know which implement will be compatible with the tractor and which functionalities will be available with each possible tractor/implement combination. The AEF Isobus Database and Isobus Check Tool system help to answer these questions as well as aid farmers in case of compatibility issues.

The Isobus Database is an online database of tractors, implements, and other agricultural equipment that is accessible by their respective manufacturers who are members along with their dealership networks. The equipment listed in this database is Isobus-certified and was published by their manufacturers. By accessing the database, the user can search through a list of implements and tractors by manufacturer, type, and model, as illustrated in Figure 1.

Once a piece of equipment is selected, such as a tractor, the database will display information about the equipment's product version, available functionalities, and Isobus compliance certifications. At this point the user can search for another piece of equipment, such as an implement, and select the one desired. The database will display the same type of information for the implement along with the information for the tractor. In addition, it will also show the combined functionalities between the selected tractor and implement. This allows the user to check the functionalities and certifications about an existing system, as well as determine which additional piece of equipment will yield the appropriate combined functionalities [3]. With this information, a dealership can better advice agricultural equipment customers on what tractor, implement, or device to purchase to ensure maximum functionality in his or her system.

Isobus Check Tool

If someone experiences compatibility issues with his or her equipment, AEF offers a tool to help mediate the situation: the Isobus Check Tool (Figure 2). Developed by Sontheim Industrie Elektronik, the Isobus Check Tool provides a mechanism for gathering important information about a specific combination of agricultural machines on the field and relaying that information to the manufacturers to help them work together to solve the problem. For example, ▷

when a farmer hooks up a new implement to his tractor, he might notice that certain functionalities are not working properly, and some might not be working at all. Traditionally, the farmer would call the dealer of one of the devices, such as the tractor dealer, to report the problem, hoping to get a fix fast since the harvest season is on its way. However, it can be very difficult for the technician to identify the source of the problem since it appears that the tractor itself is working correctly. Similarly, the implement service technician may struggle to identify the source of the problem since the implement does not appear to be out of order. The reason for this difficulty is that the compatibility problem is an issue related to the tractor-implement system as a whole and not necessarily an issue confined to a single standalone piece of equipment. Without a method for both manufacturers to work together on the issue, it can take a very long time for the problem to be fixed. With the Isobus Check Tool, such incidents can be remedied in a faster and more efficient manner.

The Isobus Check Tool is a software system that, in conjunction with a CAN interface, connects to the Isobus and runs and records a trace of important diagnostic information about the different ECUs (Electronic Control Units) available on the bus. In the example above, a service technician with this tool can connect directly to the tractor's Isobus and collect data from the tractor, implement, and any other devices on that bus. This information is neatly packaged up as an XML file in a zip folder. This folder can then be uploaded into the AEF Database by the technician, where a list of the devices on that bus will be displayed including information regarding their manufacturer, model, functionalities, certifications, and combined functionalities, as illustrated in Figure 3. This way the technician can pinpoint which devices are not compatible, create a ticket, and send it to the appropriate manufacturers along with any diagnostic information pulled from the system.

When a ticket is created in the database, engineers from the different manufacturers can work together to find the solution to the problem. The database provides them with a platform on which they can communicate and track the progress of the solution. It also stores information about the tickets and their solutions, which then serve as reference material that can be used to solve similar future compatibility issues. Essentially, the Isobus Check Tool and Isobus Database system provide the means to get the right people from the right companies to work together to address compatibility problems that may arise among equipment developed by different manufacturers. For this reason, the Isobus Check Tool is available to all manufacturers that are members of AEF at no charge, along with access and use of the Isobus Database.

The Isobus Database and Isobus Check Tool system was developed to provide support to the agricultural public for the Isobus standard. It provides a way for end-users of agricultural equipment to make smarter decisions on which devices to purchase to ensure maximum functionality. Furthermore, compatibility problems that an end-user might experience with the devices in his system can be solved in a much timelier manner due to the use of the Isobus Check Tool by the service technician. Manufacturers also benefit from the use of this database/check tool system. It allows them to track the certification status of their equipment, solve compatibility issues, and it provides a store of solutions, which is a valuable resource when solving compatibility problems. ◀

From doorman to CAN-controlled turnstile

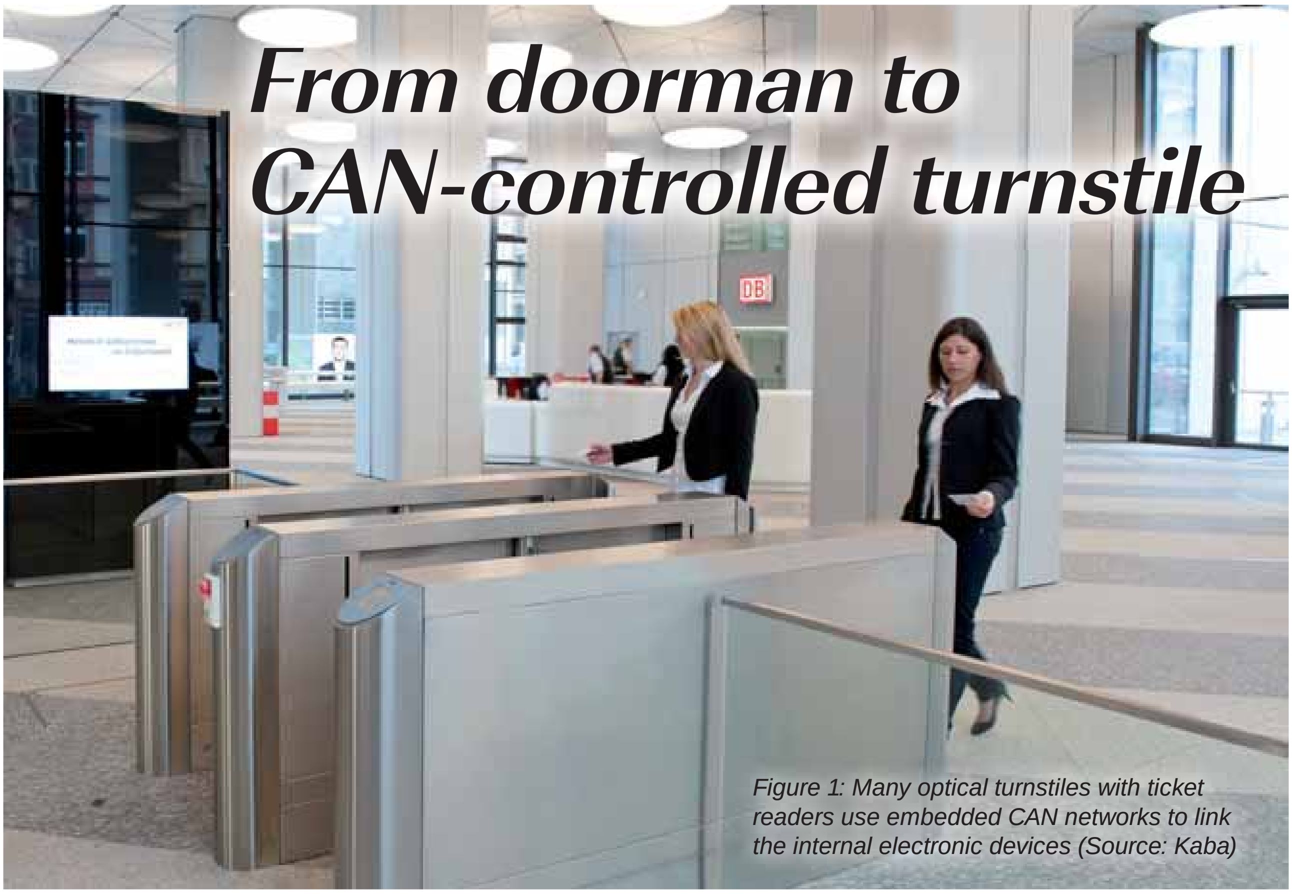


Figure 1: Many optical turnstiles with ticket readers use embedded CAN networks to link the internal electronic devices (Source: Kaba)

Doormen and gatekeepers are as old as doors and gates. In modern times, turnstiles substitute them. Some of them use embedded CAN networks.

Links
www.teamaxxess.com
www.cmolo.com
www.dresden-elektronik.de
www.a-e.cn
www.gunnebogroup.com
www.kaba.com
www.ac-magnetic.com
www.en.gdyuan.cn

Access control has a long history. During the time of the Roman playwright Plautus (245 to 184 B.C.), doorman was already an occupation. Today bouncers and security guards still supervise entrances and exits. But increasingly turnstiles are used to separate people and to control access to certain areas. It could be a gate at an airport, the entrance to a sports arena or a metro station, and many other public facilities including fairgrounds. Clarence Saunders (1881 to 1953) introduced turnstiles in his Piggly Wiggly stores. These first supermarkets allowed customers to browse the aisles and select products on their own. Shoppers entered the stores through a turnstile and followed the predetermined four-aisle path. After paying at the checkout counter, customers exited through a second turnstile.

Mechanical turnstiles often use ratchet mechanisms to allow rotation of the stile in only one direction. Modern turnstiles are often controlled electronically and are sometimes equipped with ticket readers or payment units for coins or tokens (fare-gates). They are also used to count people passing through gates.

The electronic units in a turnstile need to communicate. Often, serial links (e.g. EIA 485) are installed to exchange data between the devices. Some providers have implemented embedded CAN networks to communicate between the turnstiles and also deeply embedded CAN networks to link the motor, the sensors, and the displays to the main controller. Normally, proprietary higher-layer protocols are used. But there is an increasing need for standardized higher-layer protocols, when third-party products such as card-readers need to be integrated without re-programming them. CANopen provides all necessary functions.

Access control

Access control is one of the main purposes of turnstiles. There are many different types available: tripod turnstiles, waist- and full-height turnstiles as well as optical turnstiles – opening when a person is detected (e.g. by infrared sensors) or closing if the passing person is not authorized (no valid ticket). The first optical turnstile was developed for the San Francisco market. In most cases, they allow only one person to enter or exit. They enforce one-way traffic. ▷

CAN BE CONNECTED TO ANY FIELDBUS

750-658 CAN Gateway

750-8204

750-837

750-337

750-338

750-347

750-348

767-1501

750-658

The WAGO-I/O-SYSTEM 750 – One System for All Applications!

Gateway for all CAN protocols

CAN 2.0A, CAN 2.0B

Supports all CAN baud rates and autobaud

Operating modes: sniffer, transparent, mapped I/O

www.wago.com

WE INNOVATE!

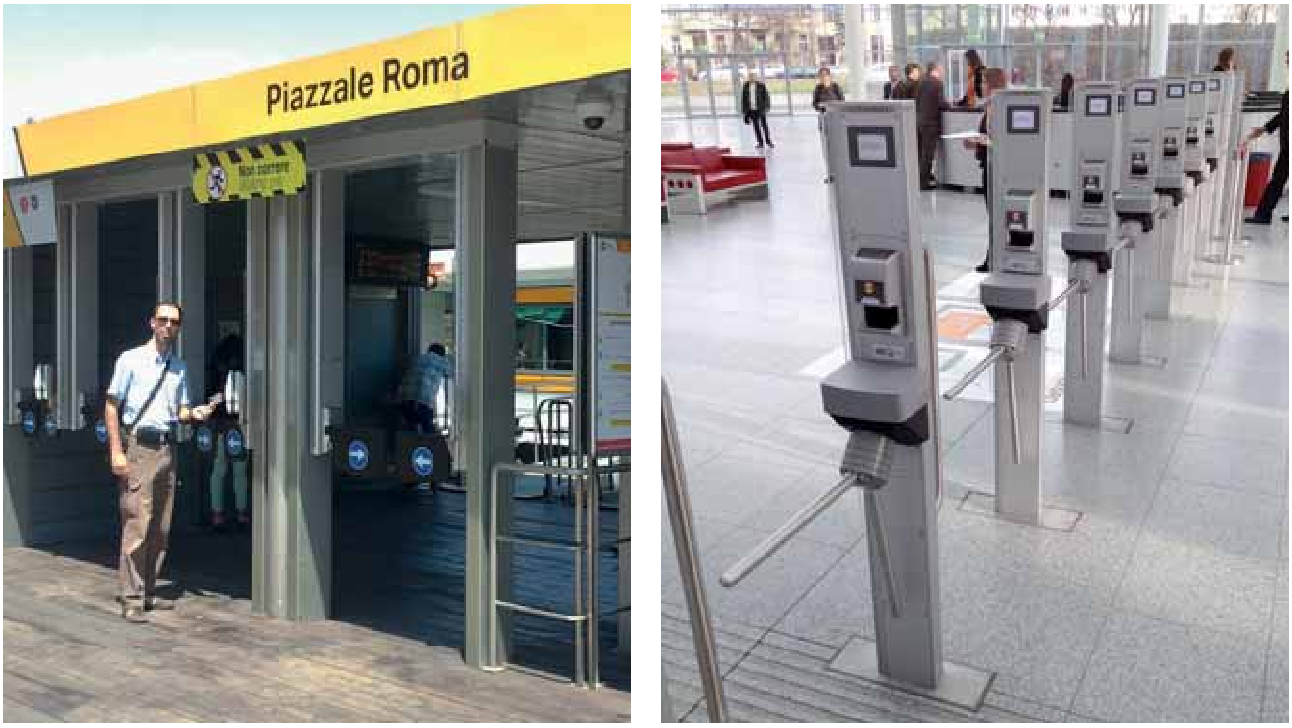


Figure 2: Typical turnstile applications at Venice's Vaporetto stations (Venice) and on the Vienna fair-ground controlling the access and counting the number of people passing through (Source: Axxess)

In the middle of the 60s, Omron in cooperation with Kinki Nippon Railway developed an automated railway station with an automatic ticket gate for commuters. Later, the Japanese company developed an automated ticket gate capable of handling both commuter passes and regular train tickets. The world's first fully automated (unmanned) train station system was completed and put into use in 1967.

Nowadays, many turnstiles are in operation all over the world and the number is growing steadily. Especially in the Far East, the turnstile business is still growing. The Pedestrian Entrance Control equipment sales surpass US-$600 million this year according to an IHS Electronic & Media's report. Of course, turnstiles are just one part of this market, which also covers speed gates, security doors, and normal entrance doors.

There are many turnstile suppliers; some operate worldwide, while others provide customized products in a specific country or application field. The Swiss company Kaba Group, founded in 1862, is one of the market leading turnstile manufacturers. The enterprise produces for example the Kerberos tripod turnstiles, half- and full-height turnstiles as well as swing, sliding, and revolving doors. All turnstile types are controlled by the ETS 21 controller, which features CAN connectivity. In 2013, the company reported a turnover of about 1 billion CHF. This figure exceeded the target. The 6,8-% growth in Asia was higher than in America (5,2 %) and Europe (4,9 %). Kaba is adjusting its group structure; it will complete the process by the end of 2014. The existing Access + Data Systems (ADS) EMEA/AP division, which currently generates around 60 % of consolidated turnover, is being split into an ADS EMEA (Europe, Middle East and Africa) division and an ADS AP (Asia Pacific) division. Just over 10 % of Kaba's turnover ▷

Dynamic traffic signs

Dresden Elektronik (Germany) has developed dynamic signposting solutions based on CAN networks. The modular system makes it possible to equip one location with up to 100 prism groups, which can be up to 500 m away from the outstation. The communication between sign and outstation works via CAN and the communication with the control center via Ethernet. The IEC 61131-3 programmable outstation provides up to ten CAN interfaces.

Dynamic signs display information depending on the current traffic situation. Bottlenecks can be detected via a control center and displayed guidance routes can be optimized. Traffic obstructions can be improved and solved without major effort. The company also offers traffic light systems, which implement up to four CAN networks.

Dynamic traffic signs are connected to the outstation via a CAN network; the outstation comprises up to ten network interfaces (Source: Dresden Elektronik)

The networks allow configuring application-specific solutions. The maximum length of the networks is about 500 m.

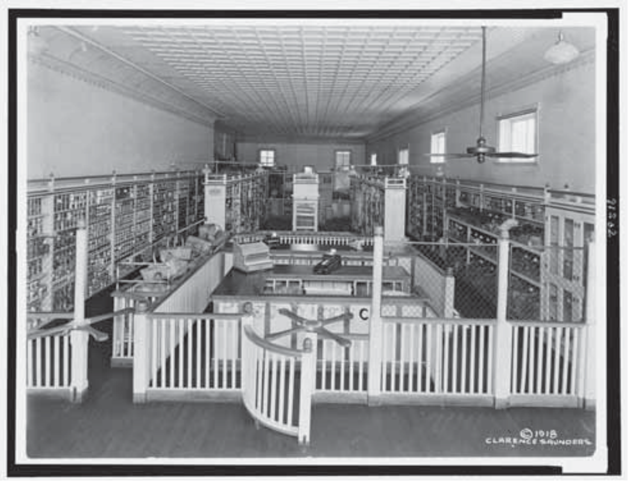


Figure 3: As early as 1918, Piggly Wiggly stores used turnstiles at the entrance and at the exit

is generated in Asia at the moment, and the aim is to increase this proportion profitably. Recently, the company has made acquisitions in China and India.

Cmolo (China) is one of the competitors in the Asian turnstile market. The company provides optional CAN connectivity for its products including embedded motor controllers. Yuan, another Chinese tripod turnstile provider, also offers an optional CAN interface. Another Chinese turnstile company, Essence, founded in 1999, also uses optional embedded CAN networks in its ES2000 and ES3000 tripod turnstiles. In many of the turnstiles, servomotors can optionally be connected to an embedded CAN network. The Smartgate by Access (Austria) comes with an internal CAN network and communicates with other turnstiles by means of wireless communication. Its AX500 CAN-connectable Linux-based control module can optionally be equipped with an operator display. The Austrian company is similarly increasingly active in Asia: In China, Axess's systems have already been in operation since the beginning of 2014. The first installation in Japan was installed recently. In early summer, a contract with one of the largest operators of Japan, which administers 21 resorts, was signed. The SXT Smartaxess terminal is connected to a unit that counts the number of people passing through the turnstile via CAN. This is another important task of modern access control systems such as turnstiles.

Magnetic Autocontrol (USA) also operates globally. The company provides all kinds of turnstile types and swing gates. Optionally MMC-12X motor controllers and MBC-110 central control unit as well as other devices communicate via CAN. The Swedish company Gunnebo offers optional CAN connectivity for its Boardsec optical turnstile, too. The Safecoin coin roll dispensing system by the same company also uses an embedded CAN network.

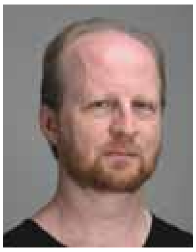
The embedded and deeply embedded CAN networks used in turnstiles and similar access control systems link mainly devices manufactured by the provider. However, for rarely needed devices and low-volume applications, the employment of third-party devices could reduce development and production costs. In this case, standardized higher-layer protocols and profiles, such as CANopen, would simplify system integration. Additionally, off-the-shelf tools could be used for system integration as well as diagnostic services.

Holger Zeltwanger

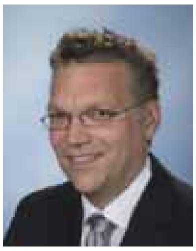
Tips and tricks for the use of CAPL (part 3)

The third and final part of this series presents tips and tricks for advanced users. Topics include associative arrays, performance, memory needs, and other database access options.

Authors



Marc Lobmeyer



Roman Marktl

Vector Informatik GmbH
Ingersheimer Str. 24
DE-70499 Stuttgart
Tel.: +49-711-80670-0
Fax: +49-711-80670-111

Link
www.vector.com

CAN Newsletter (print)
Tips and tricks for the use of CAPL (part 1)

Tips and tricks for the use of CAPL (part 2)

Unlike languages such as C, CAPL does not support any pointer objects as a reference data type and therefore has no dynamic memory management. This makes CAPL very robust, and therefore well-suited for runtime environments that are short on memory and difficult to debug. In particular, CANoe's "CAPL-on-Board" feature benefits from this; in order to improve real-time behavior, it executes programs directly on certain hardware bus interfaces. Having said that, memory is seldom in short supply in the Windows' runtime environment. Therefore in this runtime environment CAPL offers associative arrays that can be used to store data even if the amount of data to be stored is unknown at the program start. Associative arrays are containers which are equivalent to maps or dynamic arrays of other programming languages. Internally, CAPL uses an efficient hash table for these arrays. Consequently, these special arrays enable saving bus messages or measurement values, even if it is unknown in advance which messages or how many measurement values will occur.

In CAPL, associative arrays are declared as simple arrays, but with a key type instead of the otherwise usual size entry. Two examples of associative arrays:

```
long lastTime [long];
char[30] translate[ char[] ];
```

The variable *lastTime* is an array that maps *long* keys to *long* values, while *translate* maps *string* keys (without length limitation!) to *string* values up to 30 characters. The following example uses *lastTime* to store a time value for each message ID occurring on the CAN network:

```
on message CAN1.*{
    lastTime [this.id]
      = this.time;
  }
```

To enhance the user's experience, CAPL provides the following list of methods for associative array variables using dot notation:

- *ContainsKey* queries whether a specific key is already contained;
- *Size* returns the number of contained keys;
- *Remove* removes one key from the associative array;
- *Clear* fully empties an associative array.

In fact, *Remove* and *Clear* free up memory.

Finally, there is a special form of the *for* instruction for associative arrays. This form iterates over all keys actually contained in *lastTime:*

```
for (long key: lastTime)
  {[…]} …
```

Access to databases

Part 1 of this article series already illustrated the primary use of bus-specific databases in CAPL: they make it possible to introduce names for messages and signals. From a programming perspective, the complicated aspect of signals is that they are usually tightly packed in the data payload of messages for efficiency reasons. Therefore, signals generally exhibit arbitrary bit lengths and positions within the data payload of a message. They can also be stored in either Intel or Motorola format.

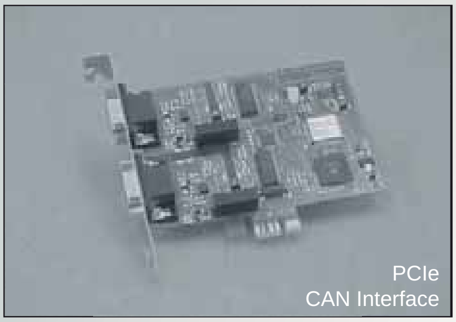
Symbol-based access via a signal name relieves the CAPL user of all of these details. In the case of reading or setting a signal value, the CAPL compiler automatically accounts for the signal's precise bit pattern that may include masking, swapping and shifting the bits.

To enhance user friendliness, other definable objects in the database may improve the linguistics of CAPL programming. For example, symbolic value tables may be associated with signals to use plain text names for signal value states. Furthermore, authors of a database have the freedom to define other attribute objects and to use them in the program code.

CAPL is able to use database objects directly based on their symbolic names. However, sometimes the potential objects of interest are not known at the time of program implementation. Therefore, the CAPL user may dynamically access the symbolic names and properties such as message names and identifiers transmitted by a network node. A brief example:

```
message * m;
int i, mx;
mx=elcount(aNet::aNode.Tx);
for (i = 0; i < mx; ++i)
{
  m.id=aNet::aNode.TX[i];
  write(DBLookup(m).Name);
}
```

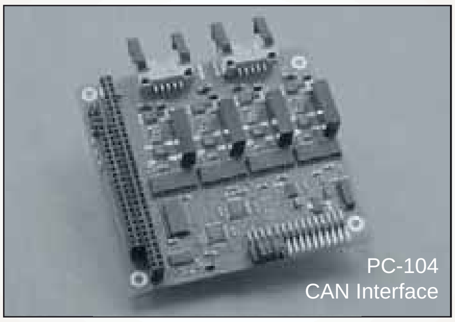
▷


PCIe
CAN Interface

USB
CAN Interface

PCI-104
CAN Interface

PCI
CAN Interface


PC-104
CAN Interface

Ethernet
CAN Interface

CAN Interfaces for Your Requirements

- Economical solutions for series applications
- Optimized for industrial applications
- Solutions for stationary and mobile use
- Software support for bus-analysis, measurement and control

www.ems-wuensche.com

EMS
Thomas Wünsche

Sonnenhang 3
D-85304 Ilmmünster
Tel. +49-8441-490260
Fax. +49-8441-81860

TWK ELEKTRONIK

Safe Switching Cam Encoders, safe rotary encoders and safe Inclinometers

With CANopen and CANopen Safety interface (SIL2 regarding IEC 61508) for engineering equipment, machinery and instrumentation

Switching Cam Encoders

Play-free electronic switching cam encoders instead of mechanical cam switches

Integrated electro-magnetic encoder

Position signal and output switches (max. 4) programmable via CANopen or CANoben safety bus system

Safe limit switches certified according to SIL2 regarding ICE 61508

Inclinometers

Contactless, wear-free sensor system in MEMS technology

1 or 2 measuring axes

Selectable measuring range

Inclination signal output according to CANopen

SIL2 regarding IEC 61508 certified for safety applications with CANopen safety interface plus special shock- and vibration procedures for mobile machines

Inductive Linear Tranducers

Measuring ranges up to 360 mm

Completely potted housing

Various mechanical designs to meet specific application requirements

Magnetic Rotary Encoders

36 to 58 mm diameter

up to 22 Bit/360°

SIL2 regarding IEC 61508 certified for safety applications with CANopen safety interface for mono- and multiturn

With bootloader functionality

CANopen

CANopen safety

www.twk.de | info@twk.de

20+ years Membership

These symbolic access methods allow the user to implement generic programs – together with the previously introduced associative arrays.

Performance

Most CAPL programs must meet non-trivial real-time conditions. The execution model of a node simulated with CAPL even follows the model concept that CAPL programs can be executed at any speed (see part 2 of this series of articles). To adequately approach this ideal, CAPL programs are compiled, i.e. they are compiled into the machine language of the specific executing microprocessor. Moreover, optimized code sequences are used for the often complex access to signals. Below are a few tips on how the user can affect performance.

writeEx(): the *write* function is used to output specific texts to the Write window in CANoe and CANalyzer. As an alternative, the *writeEx* function is available for outputting larger quantities of data. For one, it can be used to write directly to the Trace window or to a log file. The text output generated by *writeEx* is in all respects treated like a bus event, including the high priority processing and synchronizing the time stamps with real bus events.

Event procedures: a CAPL program consists of a combination of procedures that react to events. Some of these events may occur very frequently. Therefore, a program's performance is significantly better if only those events get processed, which are concerned. For example, if the user is only interested in those Flexray *slots* that contain a specific signal, it is more efficient to define on `frSlot signalname` than on `frSlot *`.

Signal edges: there are two event procedure versions for signals and system variables. *on signal_update* and *on sysvar_update* are called with each write access to the specific data objects, even if the object’s value has not changed at all. By contrast, *on signal_change* (*on signal* in short) and *on sysvar_change* (*on sysvar* in short) offer a performance advantage if only signal edges are to be handled. Those event procedures are optimized to trigger on value changes only.

Memory needs

Unlike most block-oriented languages, such as C, all locally defined variables in CAPL are static by default. This means that they are all created at the program start, and memory used to store these variables is not freed until the end of the program. Consequently, CAPL may require a surprisingly large amount of memory if many event procedures define the same type of large variables, which they could actually share. An example:

```
testcase test789()
{
  char outBuffer[1024];
  [..]
```

There are CAPL programs with thousands of such test procedures, of which only one may be executed at any given time. Rather than defining a large local variable of the same type in each event procedure, defining the large variable once globally in the Variables section utilizes a lot less memory.

Another inadvisable practice is to create very large arrays, e.g. to store event data under the respective message IDs. An extended ID in CAN comprises 29 bits, so it can assume over 500 million values. To define an array for this purpose would be a waste of memory. In such cases, it is better to use associative arrays as described above. Although associative arrays need somewhat more memory for each key that is actually used, they do not need any memory for keys that are not used.

CAPL

CAPL is a procedural programming language similar to C, which was developed by Vector Informatik. The execution of program blocks is controlled by events. CAPL programs are developed and compiled in a dedicated browser. This makes it possible to access all of the objects contained in the database (messages, signals, environment variables) as well as system variables. In addition, CAPL provides many predefined functions that support working with the CANoe and CANalyzer development, testing and simulation tools.

Useful, relatively unknown features

CAPL offers a number of less familiar and mainly newer features:

Structs can be used to define structures, similar to the approach in C. Together with copying operations, which can also convert Intel and Motorola formats within a *struct*, they represent a flexible method for data conversion.

When CAPL functions are called, the user has the option of passing reference parameters in addition to value parameters. Reference parameters make it possible to return more than one result value from a function. Reference parameters can also be used within CAPL–DLLs.

CAPL programs should also not crash in case of faulty usage. On one hand, this robustness is attained by the language structure, since there are no general pointers. On the other hand, stability is improved by automatic runtime checks of array limits, stack limits and the necessary computing time.

A separate command-line version of the compiler is available. This version is very helpful in automating sequences in script languages.

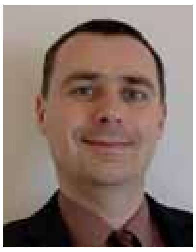
Concluding Remarks

This series of articles has introduced CAPL as an example of a problem-oriented programming language. The familiar C language syntax of CAPL simplifies the user’s learning curve. Specific symbolic databases and concepts for using CAPL in simulation, emulation, and testing of fieldbus nodes support the application domains. Vector is carefully and continually extending the language in a way that maintains compatibility with previous versions while cultivating new application areas. ◀

CANopen gateways to the Internet of Things

The CiA organization has started several activities to specify Internet access for CANopen entities. These projects could make CANopen part of the Internet of Things.

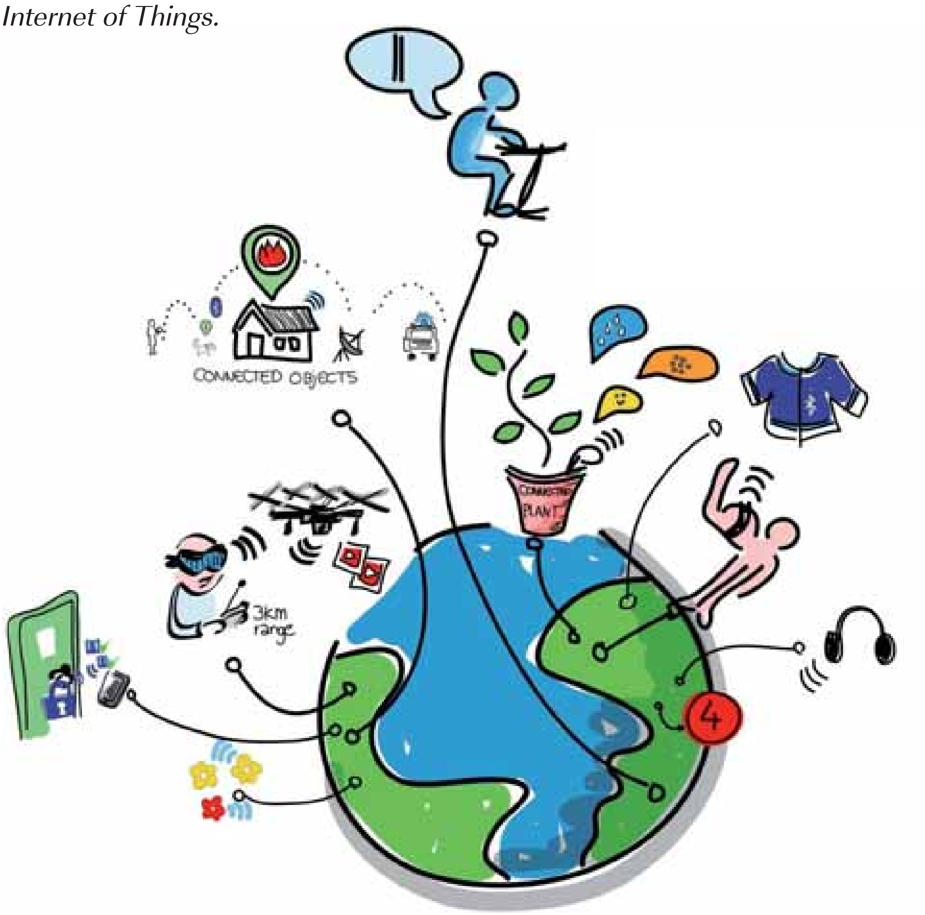
Author



Torsten Gedenk

Emtas GmbH
Fritz-Haber-Str. 9
DE-06217 Merseburg
Tel.: +49-3461-79416-18
Fax: +49-3461-79416-10
service@emtas.de

Link
www.emtas.de



(Author: Wilgengebroed, License: CC-BY 2.0)

The CiA 309 gateway specification series ("Access from other networks") is suitable for connections between CANopen and TCP/IP-based networks. Therefore it can also be used for the so-called Internet of Things (IoT). The first CiA 309 version, which was defined in 2004, was only used in niche applications but starting in 2012, when the specification was updated, CiA 309 gateways have been in use in a broad range of applications and several hardware and software products are now available. Especially with the Internet of Things more applications have opened up for CiA 309 gateways.

An overview of CiA 309

The CiA 309 specification consists of three parts. The first part, CiA 309-1, describes general services and principles and defines three gateway classes.

Besides these classes, additional CANopen services like PDO, heartbeat consumer, node guarding master, LSS master, and more are defined in CiA 309-1, but those are optional.

Table 1: CANopen gateway classes

Class 1	NMT Slave + SDO Client
Class 2	Class 1+ SDO requesting device
Class 3	CANopen Manager

The second part, CiA 309-2, defines a mapping of these services to a Modbus/TCP-CANopen gateway. The Modbus/TCP side of the gateway is a Modbus/TCP slave and the CANopen side can either be a simple SDO client or a sophisticated CANopen manager depending on the implemented gateway class. Nevertheless, the Modbus/TCP-CANopen gateway has to work within the limitations of existing Modbus

▷

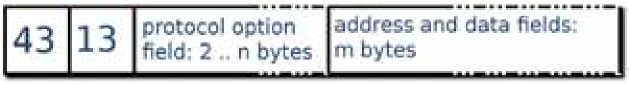

43
13
protocol option field: 2 .. n bytes
address and data fields: m bytes

Figure 1: Modbus/CANopen gateway message structure

networks, which means that the length of requests and responses is limited to 253 bytes and that asynchronous data transfers (e.g. PDOs) are not allowed. Modbus messages to Modbus/CANopen gateways are transmitted using the Modbus Encapsulated Interface Transport (MEI) with the function code 43 and the MEI type 13. The 2 bytes are followed by CiA 309-2 commands as binary data.

CiA 309-3 defines an ASCII mapping of the CANopen services and all CANopen services can be transmitted as ASCII strings via TCP/IP. Nevertheless, the protocol definition does not limit the use of TCP/IP as transport layer, so also implementation that use UDP/IP or a serial point- to-point protocol are possible and in use.

The specification basically covers four service primitives. These are:

- Request: communication service required;
- Confirmation: answer to a service request;
- Indication: an event has occurred in the network;
- Response: answer to an event.

Based on these, the ASCII protocol for CANopen defines commands that are composed of tokens that are separated by white-spaces and closed by CRLF characters. All commands that are sent to the gateway are confirmed and preceded with a sequence number that is enclosed in square brackets [ ]. The sequence number is an Unsigned32 number and this number is sent back from the gateway with the answer, but it is not used with event-triggered messages from the gateway. After the sequence number the command starts with an optional net-ID and the node-ID, which is addressed and followed by the specific command. All commands are defined in CiA 309-3 Backus–Naur Form (BNF). The definition for a SDO request is e.g.:

```
"["<sequence>"]"  [[net] node]
r[ead] <multiplexer> <datatype>
```

and an example for such a request is:

```
[2232] 1 43 r 0x1000 0 u32
```

which means that the value of the object 0x1000 subindex 0 shall be read from node 43 in net 1. If a CiA 309-3 gateway only supports a single CANopen network, the net number can be omitted.

In the last meeting of the CiA 309 working group it was decided to open the specification for more complex commands. Although nothing has been defined yet, the decision paths the way to more sophisticated use cases, which might be necessary for the Internet of Things.

Modbus/TCP and ASCII gateways

Emtas provides both a CiA 309-2 gateway to connect Modbus/TCP to CANopen networks and a CiA 309-3 gateway for TCP/IP connections using ASCII commands. The CANopen component of the gateways is based on the CANopen master stack from Emtas. The CiA 309 gateways are available as Linux applications and can be used with an (embedded) Linux device that supports a can4linux or SocketCAN interface. Also, a source code edition is offered that can be ported to all targets that support a CAN interface and a TCP/IP stack. Fully ▷

featured TCP/IP stacks with BSD sockets facilitate the use, but light-weight TCP/IP stacks without BSD socket support can also be used. Thus the source code edition is suitable for integration into small embedded devices. Additionally, using the source code it is possible to add functions and services that exceed the scope of the CiA 309 specification.

Current use cases of CiA 309

One of the first use cases of CiA 309 were CANopen service and diagnostic tools that could operate via Ethernet or Internet connections. The first product that implemented CiA 309 (specifically CiA 309-3) in hardware was Ether-CAN, developed and manufactured by the company EMS Wünsche from Germany. Besides CANopen tools, the CANopen specification 443 for subsea instruments specifies the use of CiA 309-3 for a transparent maintenance link to configure or update devices. More applications exist as backbones and configuration links to handle parameterization and firmware updates, but they haven't been specified in CANopen application profiles yet.

Use cases of Internet of Things

According to Wikipedia, the Internet of Things (IoT) refers to the interconnection of uniquely identifiable embedded computing-like devices within the existing Internet infrastructure. Typically, it is expected to offer advanced connectivity of devices, systems, and services that goes beyond machine-to-machine communication (M2M) and covers a variety of protocols, domains, and applications.

In our CANopen world, the IoT means to get Internet access to CANopen networks and even single CANopen devices.

CAN in Automation recently established a working group that deals with the Internet of Things. Employees of Emtas participate in this group. Unfortunately, these experts have just started working on the topic and consequently there are no results available yet. During the first meetings, use cases have been defined: Diagnostics of devices and functional addressing. Functional addressing means that a CANopen device is no longer addressed by its CANopen node-ID but instead by a functionality, e.g. “Temperature sensor 4” or a unique function code that represents the functionality. This will not be limited to the scope of nodes but will also cover parameters and data that are normally addressed by an index and a sub index.

Additionally, it was discussed that there should be different types of CANopen devices with different IoT capabilities:

- CANopen devices with full Ethernet capabilities that might be able to be a web server themselves;
- CANopen devices with limited IoT capabilities that can only respond to a restricted set of IoT requests;
- CANopen devices without IoT support – classic CANopen devices which have to be addressed via an intelligent gateway.

Up to now the preferred method to retrieve data via an Ethernet network is to use dedicated HTTP requests, which are tunneled via CANopen networks using existing CANopen services. For all use cases and applications, LAN and WLAN access must be enabled and security considerations have to be taken into account as well.

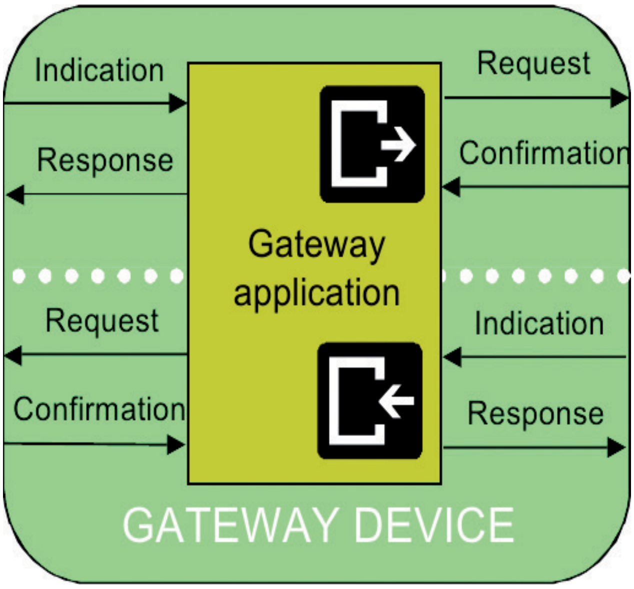


Figure 2: Gateway service primitives

CiA 309 and Energybus

The topic Internet of Things is also being discussed within Energybus, which has developed the CiA 454 application profile for use in light-electric vehicles and other energy management networks. Without ready solutions from the CANopen SIG “Internet of Things”, own discussions have been started that led to a first proposal. A current idea is to extend CiA 309 to use cases of Energybus. As discussed within the SIG, a functional addressing scheme must be added to the geographical addressing CANopen provides today. Mapped to CiA 309 this means:

- Usage of device or function names instead of node-IDs: e.g. Battery_2 instead of Node-ID 31;
- Usage of parameter names instead of index/ sub-index addressing;
- Addressing data instead of PDOs;

The usage of device names instead of node-IDs is especially important in CiA 454 (CANopen profile for energy management systems) networks. In this profile, node-IDs are usually assigned dynamically using the Layer Setting Services (LSS).

Additionally, it was proposed that a CiA 309 gateway should only transmit PDO data via TCP/IP if any values have been changed. When this is taken into account, an additional update timer is necessary to ensure that even clients that are connected later can get informed about current, but slowly-changing values. Instead, a “Request PDO Values” service could be added to CiA 309 gateways. Mapped to CiA 309-3 these new commands look like these:

```
Value Read Request similar to
SDO read request
"["<sequence>"]"  <device>
r[ead] <parameter_name>
example:
>[1234] BATTERY_2 r
rated_voltage
<[1234]] 4800

Value Write Request similar to
SDO write request
"["<sequence>"]"  <device>
w[ead] <parameter_name>
<value>
example:
>[815] MCU w assistance_level 2
<[815] OK

Value Registration
"["<sequence>"]"  <device> reg-
ister value <parameter_name>
example:
>[4711] BATTERY_2  register
value  current_voltage
<[4711] OK

Value Indication
value <parameter_name> <value>
example:
<VALUE BATTERY_2 current_
voltage 4761
```

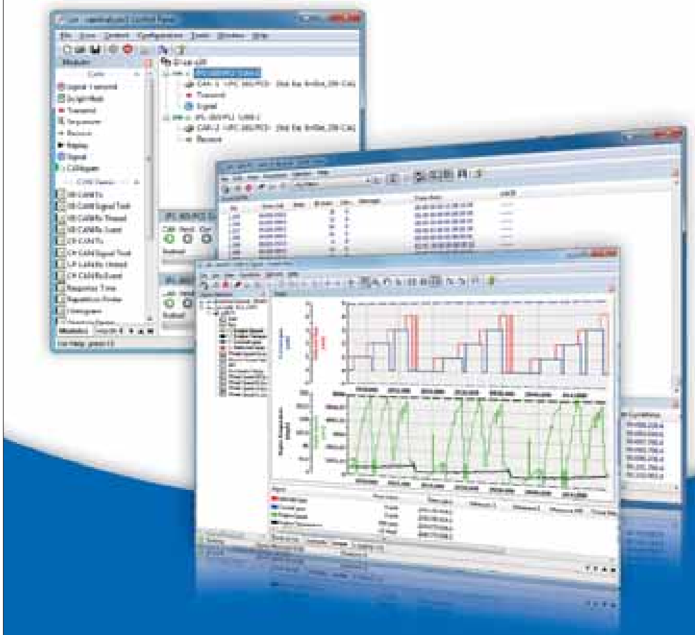
One of the objectives of that approach is for smartphone applications to be able to read/write certain information from an electric bicycle without having to know much about CANopen or Energybus (CiA 454) whereas more sophisticated applications or PC tools could use the full features of CiA 309-3.

Using such an approach, an extended CiA 309 gateway in an Energybus network has to be aware of the characteristics of CiA 454 devices and the structure of the network. Thus an extended CiA 309-3 gateway for Energybus would have to be located inside the Energybus Controller, which is the virtual device that controls the Energybus network. Besides CiA 309-3 other methods like HTTP-POST requests or more specific JSON or XMLHttpRequest requests have been considered as well. For the sake of backward compatibility these approaches were rejected for the time being in favor of extending CiA 309-3. Nevertheless this decision could be reversed if the SIG “Internet of Things” comes up with a better generic approach that fits the needs of most CANopen users.

The proposed idea to extend CiA 309 will suit the needs of the author for the use case in Energybus (CiA 454) networks. On the one hand, the CiA 454 approach could be used for a broader range of applications, but significantly more examinations have to be done beforehand. On the other hand, the work started by the CiA working group shows promising approaches. Interested parties are welcome to join the efforts to develop a generic solution for the Internet of Things even beyond the current scope of CANopen. ◀

References

- CiA 309: Access from other networks, Nuremberg, 2014
- CiA 454: Application profile for energy management systems, Nuremberg, 2014
- Holger Zeltwanger: Gateway profiles connecting CANopen and Ethernet, CAN in Automation e. V., Proceedings of the international CAN Conference 2005



Detecting potential differences

Communication disturbances attributable to potential differences in CAN units have often been underestimated. They usually go unnoticed. Such errors can nonetheless be detected, measured, and rectified.

Authors

Hendrik Stephani

Antje Wappler

Gemac mbH
Zwickauer Straße 227
DE-09116 Chemnitz
Tel.: +49-371-3377-0
Fax: +49-371-3377-272
info@gemac-chemnitz.de

Link
www.gemac-chemnitz.de

Serial bus systems are a decisive factor for determining performance capabilities of complex manufacturing systems in many industries. The whole electronic communication is realized within complex systems, meaning that the highest demands must be placed on the reliable functioning of serial bus systems. Measuring devices for bus analysis – both at the time of installation and for permanent status monitoring and early error detection – are in the meantime indispensable. On the other hand, such devices have to date remained oblivious of disturbances in data communication. These disturbances result from inadequate potential equalizations.

Until a few years ago, it was assumed that such problems were caused by system-internal reasons. Today, however, we know that external influences such as electromagnetic interference or inadequate potential equalization are increasingly the culprits where communication is disturbed. Outdated or inappropriate framework conditions (e.g. grounding and potential equalization) also open the door wider to previously ▷

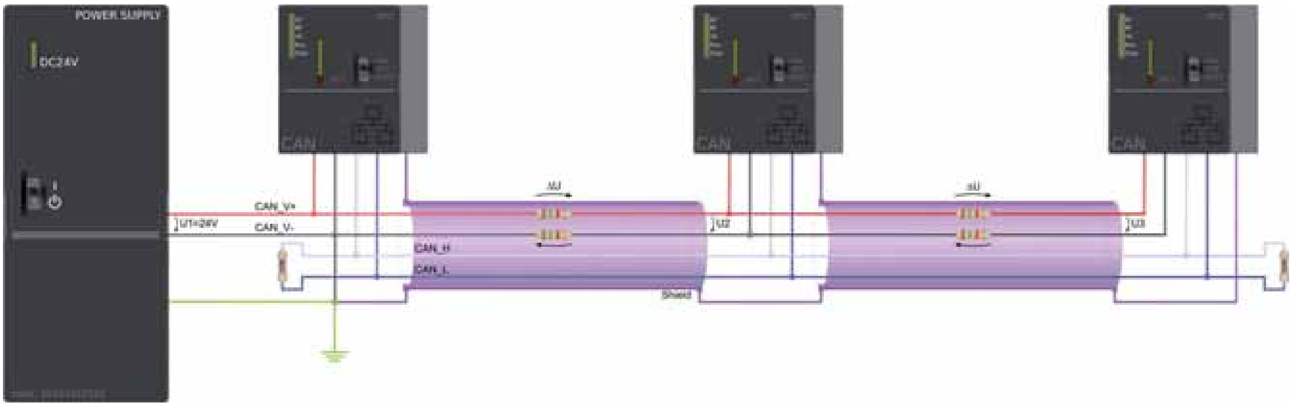


Figure 1: Wiring variant 1; all nodes supplied via CAN cable

ignored sources of disturbance. High-frequency currents, for example, often use the shielding of a data line as their return path, even when a potential equalization conductor is provided precisely for this purpose. That results in correspondingly error-prone communication, or even loss of the whole system functionality. Gemac has applied this knowledge in its developments: the latest CANtouch diagnosis device now also detects such error sources – in addition to the established measurements of physical bus characteristics.

Bus diagnosis

CAN uses a differential signal to compensate the influences of external interferences. In other words, the useful data signal is transmitted on two lines, which are inverted to each other (CAN_H and CAN_L). The difference between these two lines generates the signal received by each CAN transceiver. Disturbances on the bus can prevent correct detection of the bit stream. Gemac's diagnosis systems permit evaluation of the differential signal in the form of a general quality value, of the disturbance-free voltage range, and of the edge steepness.

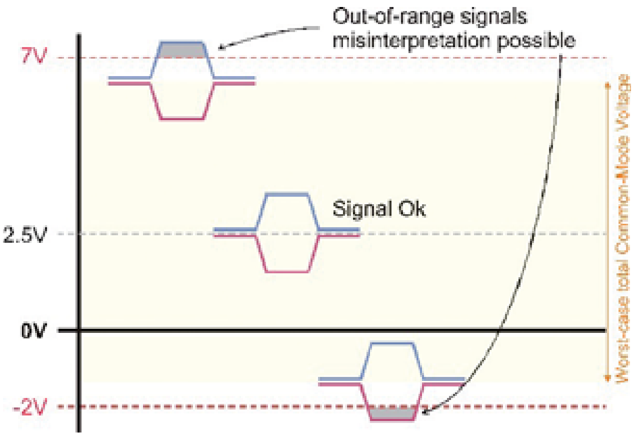

Out-of-range signals
misinterpretation possible
7V
Signal Ok
2.5V
0V
-2V
Worst-case total Common-Mode Voltage

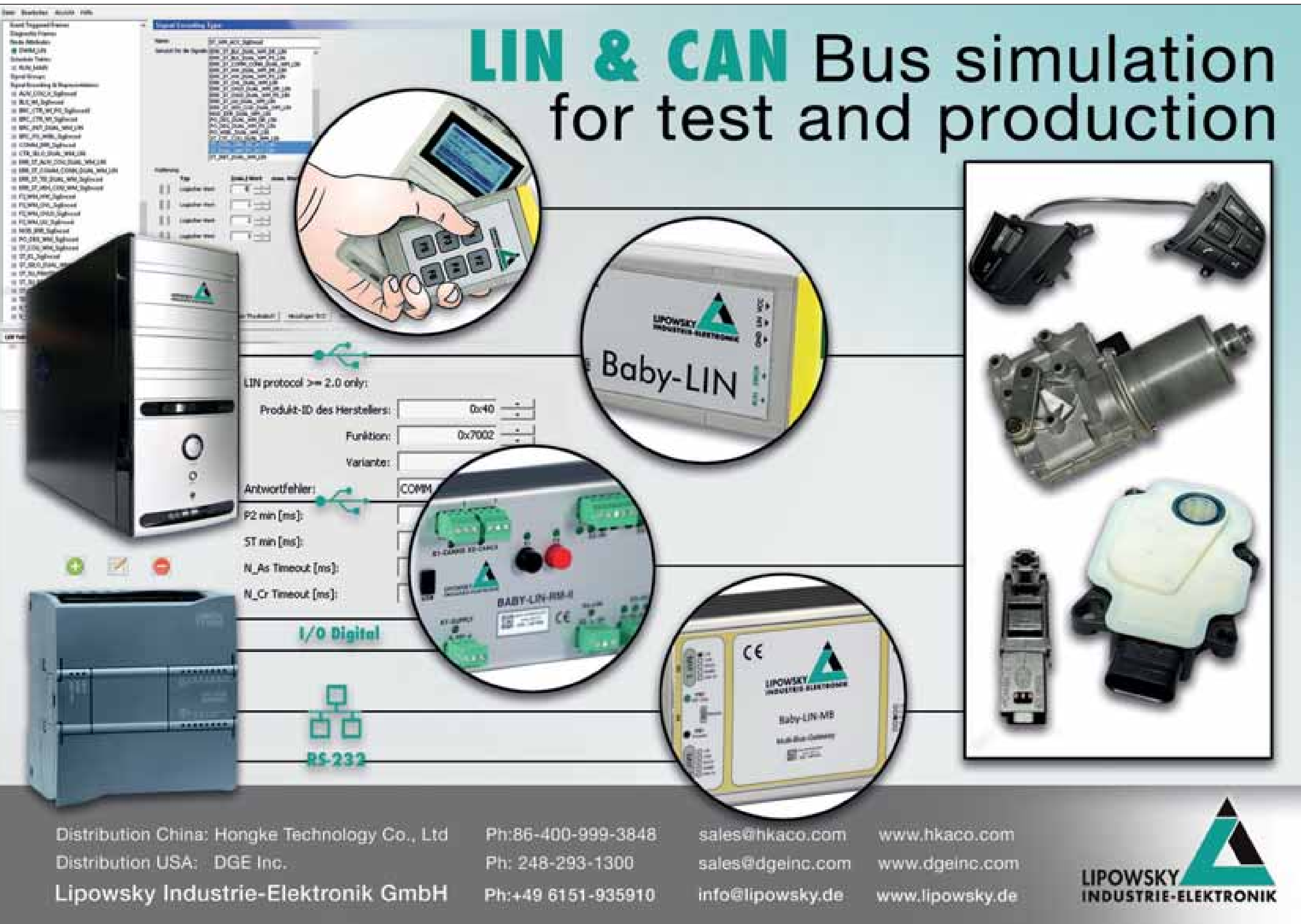
Figure 2: Communication errors in case of potential differences

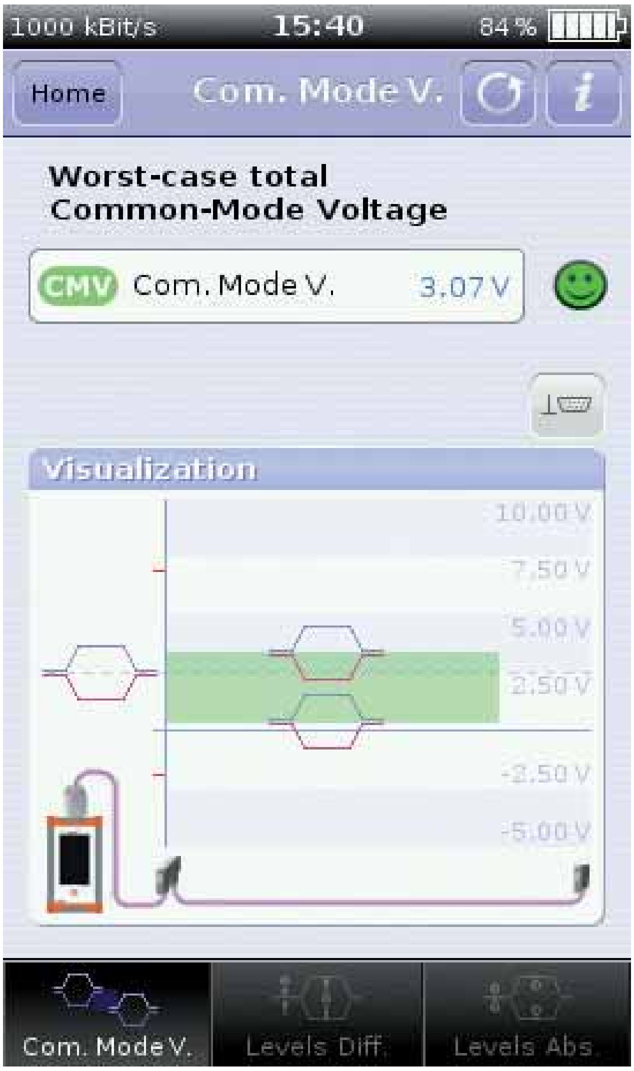
CANtouch provides absolute measurement of the individual signals CAN_H and CAN_L against a reference potential. This lets it detect an error source which is frequently encountered in system installations: the so-called "common-mode voltage". In a differentially operating transfer system, such as CAN, the term "common-mode voltage" is used for the voltage of both signals relative to a common reference potential. This is normally CAN_GND, which in each device is connected to CAN_V-. On a CAN network, both signal lines (CAN_H and CAN_L) should display a common-mode voltage of 2,5 V in the recessive state. For a number of reasons, the common-mode voltage of the devices may manifest an offset. CANtouch is able to determine this voltage offset directly. It could also be detected indirectly via measurement of the shield voltage.

In practice, two forms of wiring can lead to potential differences between devices. In the first variant, all bus nodes receive their power supply via the CAN cable; in the second, each device possesses its own power supply.

Potential confusion

Let us first consider the case where all connected devices are supplied via the four-wire CAN cable (Figure 1).





1000 kBit/s 15:40 84 %

Worst-case total Common-Mode Voltage

CMV Com. Mode V. 3.07 V

Figure 3: Measurement of the "worst-case total common-mode voltage" with smileys to assist result assessment

Two of the four wires in the cable are used for the CAN communication, and the remaining two for the supply voltage. When the wiring is installed, the shield initially has no low-resistance connection to a particular potential, as the connection to V- in each device is usually realized by way of a parallel resistor (1 MΩ) and capacitor (10 nF). For a low-resistance connection, the shield should be connected with V- and the protective ground at the central voltage supply. This has the following effect:

Due to the line resistance, the current load of the individual CAN devices results in a voltage drop (ΔU) on the supply lines. This raises the voltage level of CAN_V- at each CAN device and leads to a negative voltage offset of the shield voltage measured against CAN_V-. This "normal" shield voltage should lie in the range from approximately 0 V to -4 V. CANtouch reports greater shield voltages or a shield which is not connected to CAN_V- as errors.

The voltage drop in the cable will at the same time result in different GND potentials for the CAN transceivers. This is manifested as a shift in the levels of the signal voltages, which each CAN transceiver "sees" for itself. The CAN system only permits shifts within the range of -2 V to +7 V.

The CAN transceivers expect the signal voltages to lie within this range. Even though newer circuits tolerate a wider range of -7 V to +12 V, exceeding the expected range may lead to communication errors and, in an extreme case, eventually also to the destruction of the transceiver (Figure 2). CANtouch thus determines the maximum voltage offset among all bus nodes – the so-called "worst-case total common-mode voltage" – and issues a warning if the limit values are exceeded (Figure 3). In addition, a graphical visualization shows whether the voltage offset lies above or below the present connection position. All absolute measurements are performed relative to V- on the D-Sub 9 connector (Pin 6). CANtouch even offers the possibility to switch the reference ground to an integrated 4 mm socket. This enables the individual ground potentials of the CAN devices to be measured without needing to reconnect the test device, and thus potential differences to be detected faster. A simplified evaluation system using a combination of traffic lights and smileys assists the user.

Resisting temptation

It is important to resist the temptation to connect the shield to CAN_V- on all devices, because the operating current of the devices would then flow back via the shield, as it has a lower resistance than CAN_V-. The coupling of interferences ▷

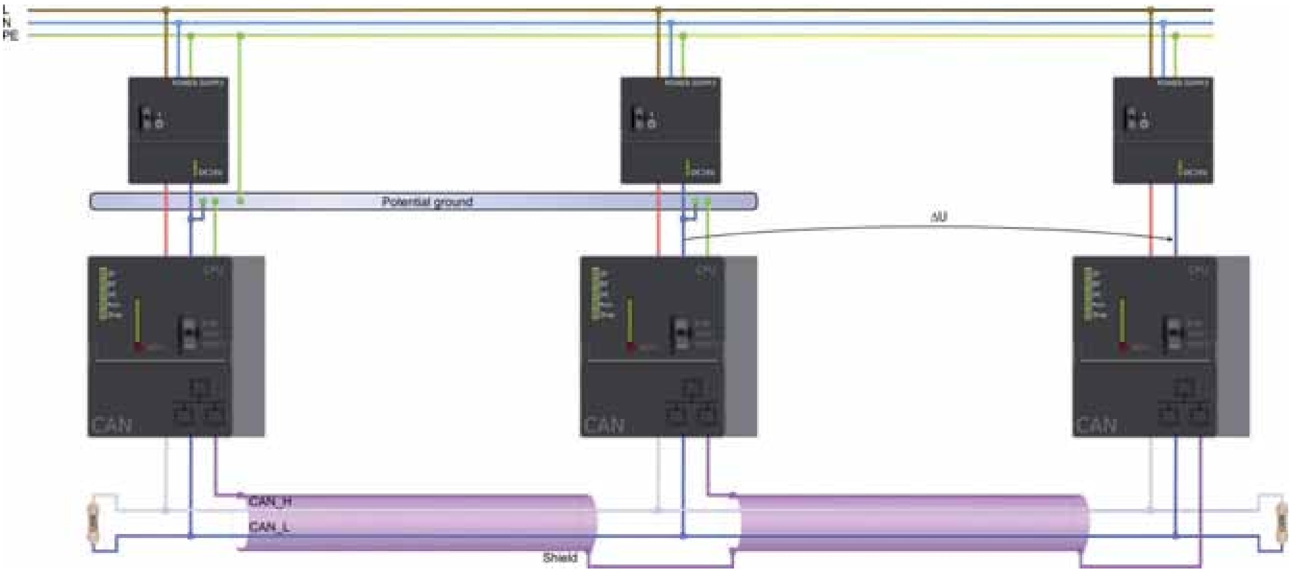


Figure 4: Separate power supply for individual bus nodes

into the signal lines is then almost inevitable. As a remedy, the voltage supply can be realized to the middle of the network, or else a supply with several power supplies can be provided. The use of CAN cables with a lower loop resistance for CAN_V+ and CAN_V- is another possibility. The CANtouch wiring test is able to measure the loop resistance of the cable.

The right cable

Standard limit values are reached relatively quickly in practice, as shown by the following example:

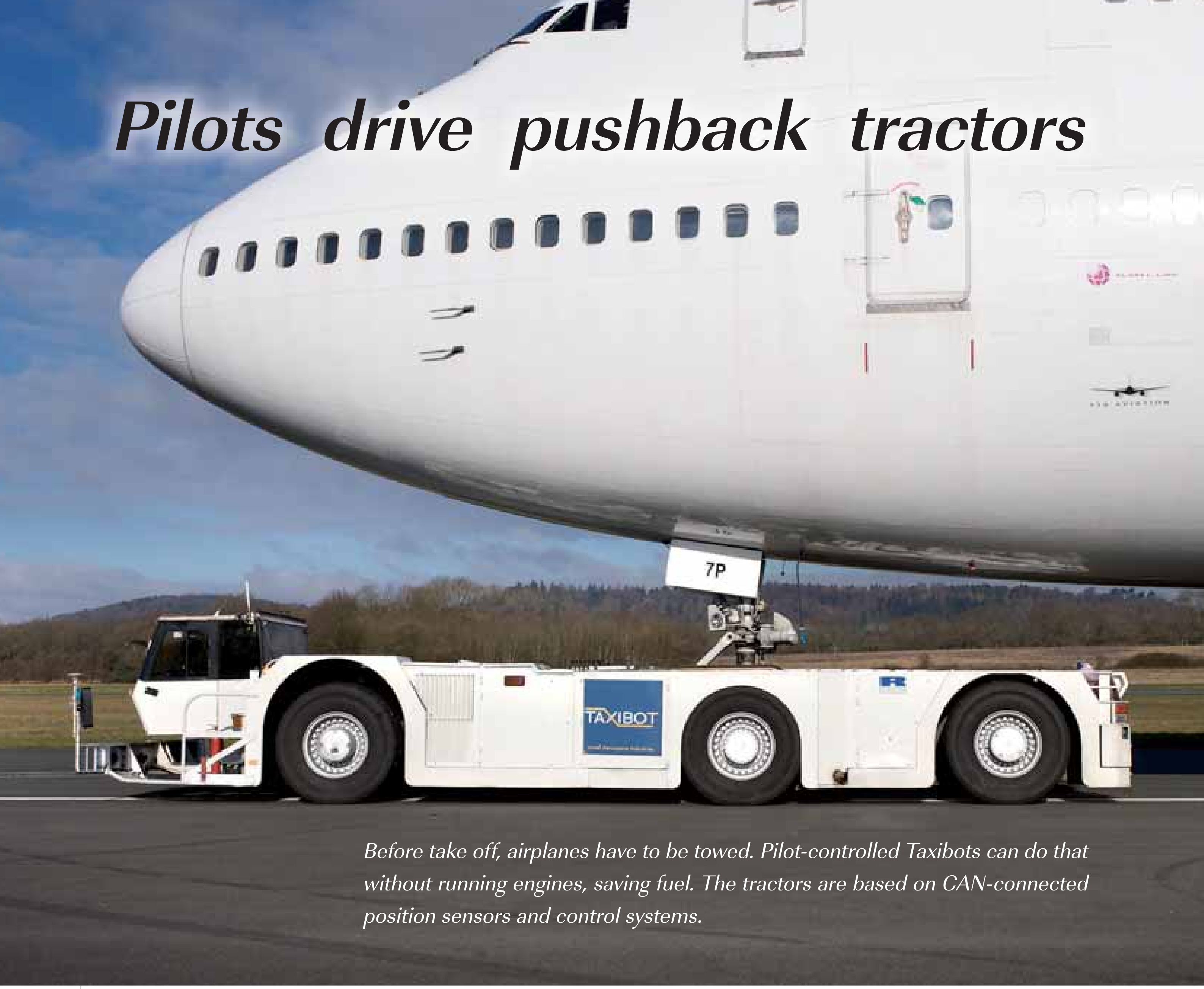
To remain within the range of -2 V to +7 V as defined by ISO 11898-2, a symmetrical potential difference of maximum 4,5 V above and below 2,5 V is permissible (2,5 V - 4,5 V = -2 V and 2,5 V + 4,5 V = +7 V). Taking a typical CAN cable with a cross-section of 0,22 mm^2, a line resistance of 186 Ω/km, and an assumed total current load of 100 mA for all devices, the permissible potential difference is already reached at a line length of approximately 240 m (and at a current of 1 A already after 24 m). An improvement can be achieved by choosing a CAN cable with a larger cross-section. At a cross-section of 0,34 mm^2, the loop resistance is reduced to 115 Ω/km, at 0,50 mm^2 to 78 Ω/km, and at 0,75 mm^2 to just 52 Ω/km.

A false shield is no solution

In conjunction with more extensive installations, it is not uncommon to find cabling configurations which provide an individual power supply to each bus node (Figure 4). In most cases, this is realized with a two-wire CAN cable. Here, too, there is a risk of potential differences if the devices are not interconnected to provide potential equalization. In practice, the shield is frequently abused for the purposes of potential equalization. The equalizing current which flows through the shield, however, is itself already a possible source of disturbances for the CAN communication and should thus be eliminated as a possible solution from the very beginning. CANtouch is able to spot such wiring problems by way of the – now possible – measurements of the shield and common-mode voltages.

The handheld diagnosis device is the industry-compatible equivalent to a smartphone. For the first time, system operators, technicians and developers are in a position to perform physical and logical bus analysis via an intuitive touchscreen. The device with a 4,3-inch color display is ready for immediate mobile use without an additional PC. Similar to a smartphone, the user takes his diagnosis device directly to the CAN system, connects it with the cable and immediately receives reliable measurement results – without stopping the system. The individual measuring functions are operated interactively and dynamically by way of applications (“apps”) based on finger gesture control. A simplified evaluation system using a combination of traffic lights and smileys assists the user in assessment of the measurement results. The CANtouch will be shown at the company's booth at the SPS IPC Drives 2014 in Nuremberg.

Pilots drive pushback tractors



Before take off, airplanes have to be towed. Pilot-controlled Taxibots can do that without running engines, saving fuel. The tractors are based on CAN-connected position sensors and control systems.

Links
www.dintec.fr
www.iai.co.il
www.sensor-technik.de

Related article
Separating non-safety and safety software functions in ECUs

Airplanes taxiing on taxiways in airports burn a large amount of fuel, emit tons of CO_2, and are very noisy. Israel Aerospace Industry (IAI) developed a towbarless towing tractor – the Taxibot (Taxiing Robot). It is a semi-autonomous vehicle that enables airplane taxiing without engines running, controlled by the pilot, and without shortening nose landing gear life-time. Because the vehicle does not use the airplane's power resources for taxiing, the fuel consumption is reduced as well as the CO_2 emissions. According to the manufacturer, the Taxibot reduces cases of FOD (foreign object damage) by 50 percent, and decreases noise and gas pollution.

The vehicle developed by IAI provides the required power to move the airplane, without the need to change or replace the airplane's APU (auxiliary power unit). It allows the pilot full control of the system. The system uses the airplane's tiller and brake pedals like in regular taxiing. Pilot training is therefore minimal.

Dintec has developed the steering-by-wire subsystem implemented in the Taxibot tractors. The airport tractor is available in two versions: Narrow-Body (NB) and Wide-Body (WB). This summer, the pilot-controlled vehicle completed the certification tests at Frankfurt Airport. The tests were conducted with a Lufthansa B737 airplane in accordance with the official EASA and CAAI flight governing authorities. The system also works with Airbus 320 aircrafts.

Steering-by-wire

In order to fulfill safety requirements, the French sub-contractor of the steering-by-wire system selected the ESX 3XM controller by STW (Sensor-Technik Wiedemann). The controller provides six CAN interfaces and runs different higher-layer protocols (CANopen, J1939, and a proprietary safety protocol). The towbarless tractor uses different steering angles for different airplanes.

All participants of the steering-by-wire system are connected to two CAN networks: “Primary CAN” and

▷

“Emergency CAN”. Both run at 500 kbit/s. The utilized proprietary safety protocol is based on a single CAN message transmitted periodically. It contains up to 3 byte safety data. In the very same frame another 3 byte contain the bit-wise inverted safety data. Additionally, these 6 byte are protected by a CRC (cyclic redundancy check), which is also transmitted in the very same CAN frame. The German TÜV has approved this ECU's protocol for SIL-2 (Safety Integrity Level) according to IEC 61508. The protocol runs on two CAN networks. The multiple CAN network approach increases the availability of the steering-by-wire system. The master controller is redundant and also communicates via CAN with the wheel-controllers. All controllers also use one other local network to connect CANopen Safety sensors measuring the wheel position.

The steering-by-wire system by Dintec implements a triplicated steering-wheel sensor, a redundant force feedback actuator and redundant actuators with cross monitoring in each wheel of the tractor. The controller features a 10-ms main-loop and 5-ms sub-loops. It also controls the suspension.

At the Mobiltron 2014 seminar organized by STW, Anthony Dollet from the Dintec group explained the steering-by-wire system in detail: “We were looking for a scalable hardware to implement master and slave controllers with the same software and on-line configuration.” The software running on the ESX 3XM controllers was programmed in C, in order to reuse existing software parts and already existing code generation tools.

Holger Zeltwanger

Ship equipment in construction machines

Open boat bridges pose high requirements for equipment: the constant contact with seawater creates a tough environment. These requirements can be useful in other rough areas, for example construction machines.

Links
www.blinkmarine.com
www.limitor.de

Riccardo Arienti
CEO
Blink Marine

Open boat bridges pose high requirements for equipment: the constant contact with seawater creates a tough environment. These requirements can be useful in other rough areas, for example construction machines.

The PowerKey Pro (PKP), a digital keypad designed and developed by Blink Marine, is debuting in the market for agricultural and construction machines. The keypad was originally developed for the nautical sector. We met with Blink Marine's CEO Riccardo Arienti, who oversaw the move from the nautical sector into general machinery.

Q A crane is significantly different from a motorboat. Can you point out anything they have in common?

A Well, they both come from areas where product quality is a determining factor. When we first started to show the PKP to important companies in the automotive sector, we realized that most of their requests were defined in terms of robustness, in other words resistance to water, dust, atmospheric agents, UV rays, and things like that. That was a good sign right from the start. We said to ourselves: If this is what people are looking for, then our product will be a success.

It's an area where the PKP excels. It was designed for installation on an open boat bridge, therefore subjected not only to rain, sun, and UV rays, but to the devastating effects of saltwater and sea spray 365 days a year. I don't know if you've ever seen what these kinds of elements can do to switches or certain pushbuttons in just a year's time. In any case, it's hard to think of conditions that are a tougher test for the resistance of an electronic device. When it comes to robustness, even the IP67 certificate isn't necessarily the last word.

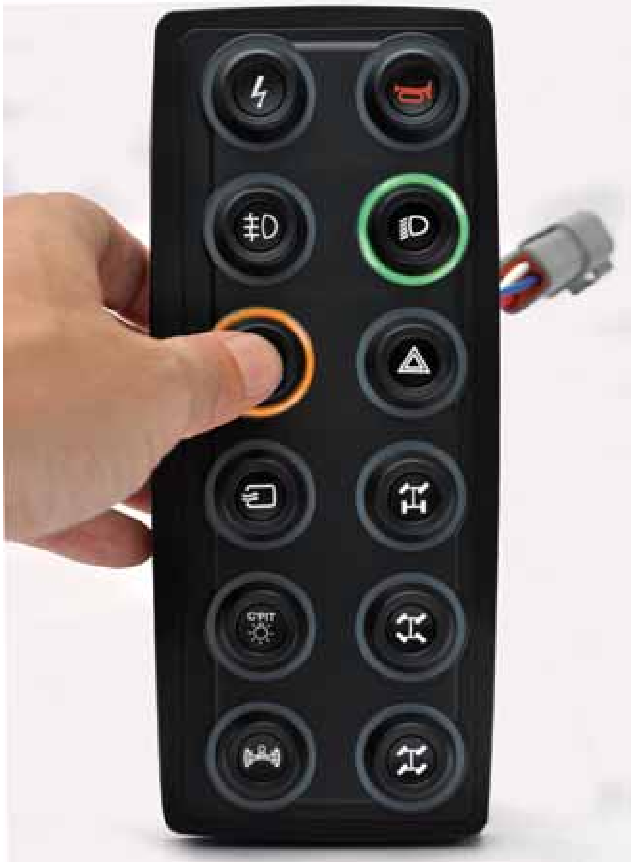


Figure 1: CAN, CANopen, J1939, Isobus, and other protocols are available with all PKP modules

Q What does IP67 not cover?

A The IP67 certificate covers resistance to the infiltration of water and solid bodies – but the PKP also possesses other characteristics that are at least as important, like the fact that it can work for a long time even in extreme temperatures (for example 24 hours at -40 °C or +85 °C) and is resistant to chemical agents.

Q What other aspects were people you spoke to interested in?

A While the first question was always about robustness, I'd have to say the second question was ▷

almost always about versatility. That is an area in which Blink Marine made ample experience while working in the nautical sector. In the nautical sector, you often have to deal with extremely specialized production realities, where people create complex, expensive machinery with no more than a handful of display products built per year. For these kinds of companies, we offer a keypad that can be transformed into a number of variations.

Q How is that possible?

A When we started working with our American partners at Digital Switching Systems, studying the product, we wondered if there wasn't some way of fixing a problem that our clients were bringing up all the time: When somebody builds a boat, they are usually able to "tailor fit" the final product to a range of client requests. This can often present a problem for purchasing keypads, since these custom requests are connected with a very small range of products, and often they were forced to purchase more keypads than they really needed just so that they could get the product they wanted on the boat they were building. To make matters worse, any request that fell outside the "standard" implied extra costs (for the client) and extra time (for us), both of which almost always appeared out of proportion with what were often minor modifications. With the PKP, we wanted to put all these issues behind us. That's why we created a system with removable inserts that make it possible to substitute any single button at any time. This way there is no single standard: the configuration of each individual keypad can be changed at any time without having to substitute the entire product. We already have more than 250 different inserts available, and we can create additional, new inserts at cost.

Q You still have the issue of minimum order quantities though.

A We made a daring decision for minimum order quantities too: there is no minimum order quantity for the PKP. We'll even accept an order for a single piece. Given the efforts we put into reaching out to clients who need top quality even for a small number of pieces, it would have been a contradiction in terms to do anything else.

Q What were the most difficult technological challenges you had to face?

A We had to adapt our keypad to the most common standards used for commercial vehicles. The first were the J1939 and NMEA2000 protocols, the most commonly used in the US. After that we had to handle requests from European producers and installers, extending the range to include CANopen protocols.

Q What else is in store for Blink Marine?

A We will promote our wired remote control for trucks in November 2014. It is suitable for using the tilt function, ECAS-, EDS- or ELM Systems and so on. We are also adding different LED light options. Last but not least, we are developing several products that will join our keypads: user interface models, as well as power management solutions for electric loads. Our aim is to make it possible for clients to rely on Blink Marine not only for keypads, but for their entire on-board systems as well. ◄

Detecting and counting unwanted particles

Condition monitoring of working fluids usually comes with a lot of drawbacks. The FCS100 series overcomes these problems with a redesigned flow-cell and traceable field calibration by the user.

Author

Bernd Donner

Elmetric GmbH
Zum Schacht 7
DE-66287 Göttelborn
Tel.: +49-6825-80185-0
info@elmetric.com

Link
www.elmetric.com

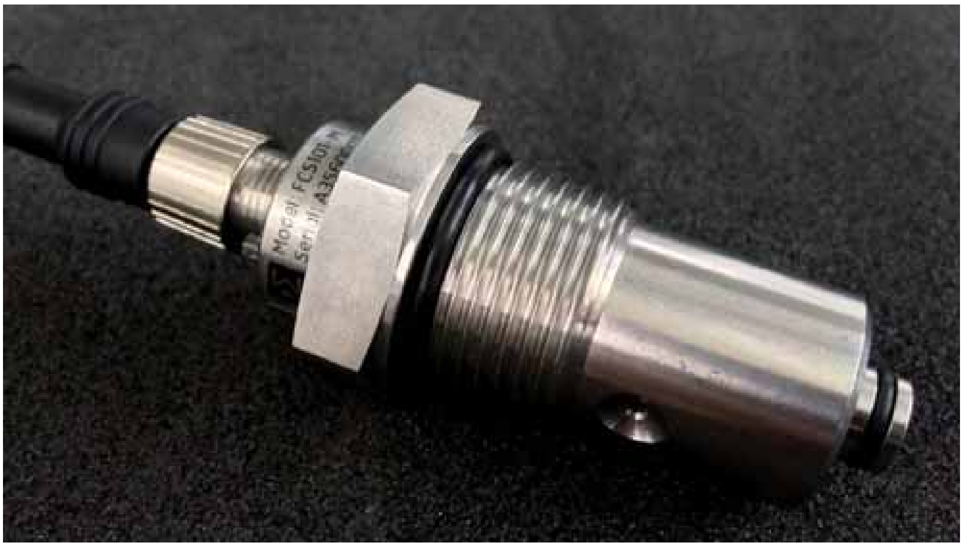


Figure 1: The FCS100 connector complies with CiA 303-1

In the field of condition monitoring of working fluids (especially hydraulic oils and lubricants) many solutions have already been developed that reliably measure and also show important fluid parameters in certain areas. Although the available devices basically do their job, they could not yet really establish themselves in a wider application front.

The reason for this is complex, but some major drawbacks are the poor integration capacity of these devices (fluidic, mechanical, and electronic), their low range, and their poor resistance to harsh environmental conditions (temperature, vibration, humidity, pressure, etc.). In addition, they are large and expensive, and the calibration of the sensors as part of an effective quality management is unsatisfactory. Consequently, the requirements for a sensor device that can be used in all areas of mobile hydraulics include a miniaturized thread-design, integration without additional fluid conditioning (fluid flow regulator), compatibility compared to all fluids without additional variants, no additional electronics, a low power consumption, no need for a mechanical adapter, and an extended measuring range. Online calculations of a system-specific risk measure should also be possible. In addition to the mastery of all standards, the additional distinction of material composition (solid, gas, ferrous metal, nonferrous metal) should be given. Other desirable characteristics are a wide operating temperature range and insensitivity to moisture and water splash. The sensors should also cover all practical pressure ranges and be vibration-resistant. These are obviously ▷

> *“The measuring range extension amounts to more than 5+ ISO classes. This makes the sensors applicable for highly contaminated fluids.*

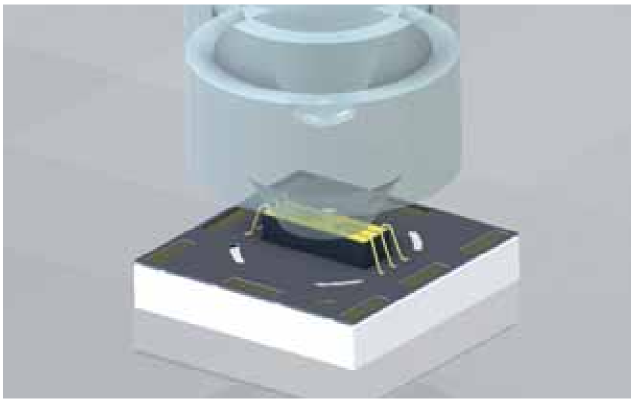


Figure 2: The core of the measuring cell – a Flip-chip semiconductor stack with triplet lens

competing parameters and there seemed to be no solution – at least not with previous designs.

Prior designs

The sensors usually consist of a flow-cell comprised of two plane-parallel glass plates, which are held apart by a metal structure. A light source – typically a laser diode – forms a "light curtain", which is perpendicular to the direction of the flow in the measuring flow-cell. Passing through the measuring cell, the light is collected from a point on the optical axis of the photodetector and converted into an analog electrical signal, which is processed and subsequently evaluated.

This design entails disadvantages that hinder miniaturization. First of all, this composition requires a great distance between light source and photodetector. Accordingly elastic internal seals are required, which limit the fluid compatibility. Thick-walled glass plates, which would make it possible to achieve high compressive strengths, cannot be used. The metallic structure, which forms the measuring channel, is fluidically less than optimal and can easily clog.

Due to the design of the measuring cell neither coincident magnetic coils nor scattered light detectors can be attached, with which the distinction between solids, ferrous or non-ferrous metals, and gas particles would be possible.

Design of the measuring cell

Key points for the design of the new measuring flow-cell were the simultaneous matching of demands for a pressure stable flow-cell, the small distance between the light source and the photodetector, and the coincident arrangement of a differential magnetic coil configuration and a detector for scattered light. The fluidic cell is represented in this design with a streamlined and extremely pressure-resistant cylindrical glass capillary. The coil system is disposed coaxially around the capillary, with a gap so that the light can pass through.

Contrary to commonly used laser diodes, a compact stack structure was used in this design. It consists of both custom AlN thinfilm substrate and a customized line-shaped LED as flip-chip construction. This stack can be fitted directly onto the circuit board in the automatic pick and place process and soldered using standard reflow, which reduces manufacturing costs.

To bridge the relatively large light path through the capillary, a new triplet lens was constructed. Through direct optical bonding between the lens and the light source or the detector and the capillary, reflection losses and at the same optical distortion through the cylindrical capillary were largely avoided. In this case, the light emitted from the LED is initially formed with a diverging lens to a reduced intermediate image of the upright. The middle lens was designed as a meniscus and provides for the shift of the main level of the overall configuration toward capillary. This made it possible to reduce the magnification to improve the optical resolution and still fulfill the law of imagery. The light transition to the capillary is formed by an optically bonded condenser. This minimizes reflection losses and ensures a high light intensity. Because in an optical imaging system, the image and the object can replace each other, it was possible to use the same structure for the receiving lenses. Thus investment in optical molding tools could be reduced. Through the use of the same parts, material and installation costs were also reduced. The ▷

lenses all have an aspherical shape and are manufactured with injection-molded plastic.

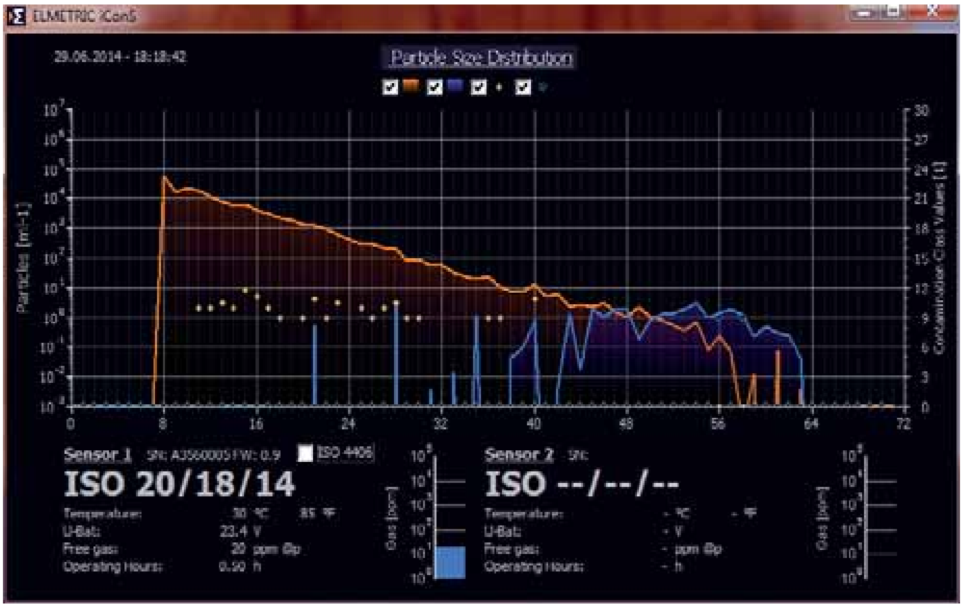
This optical construction created an intensive line-shaped light field inside the capillary through which the fluid flows. This light field interacts with passing particles as follows: 100 % of large particles are detected – these large particles are also always in low concentration in the fluid. Particles are detected less the smaller they are. The smaller they are, the higher is their natural concentration in the fluid. The relation of measurement signal to particle size is linear, resulting in a very large size range from 1 µm to 500 µm. Due to the size-dependent control of the detectability for the coincidence-free metering, the maximal measurable particle concentration could be greatly expanded. The measuring range extension amounts to more than 5+ ISO classes. This makes the sensors applicable for highly contaminated fluids.

Perpendicular and coincident to the optical axis, a light scattering detector is arranged, which also synchronously detects the signal of the scattered light to the signal of the extinction sensor. The synchronous acquisition and analysis allows reliable differentiation of solid particles and gas bubbles.

Two essential features allow reliable differentiation: Firstly, solid particles have an irregular and rough surface structure and scatter light, while gas bubbles are perfect spheres and have optically smooth surfaces, which have geometrically reproducible light scattering properties. Secondly, solid particles have either a high refractive index and/or absorb light, while gas bubbles have a refractive index of uniformly 1, which is always lower than that of the fluid. This always leads to scattered light signals, which is on the one hand proportionally to the particle size and thus also proportional to the extinction signal. On the other hand, the scattered light signal in the case of the gas bubbles is always significantly larger than the scattered light signal of solid particles. Due to the perfect geometric form of gas bubbles, their volume can be calculated. The volume fraction of free gas in the fluid is calculated and output from the sensor in ppm (parts per million). The measuring range is up to 10000 ppm with a resolution of 1 ppm.

Ease of use

Not only the cost of the sensor itself determines usability, but also the total effort required to integrate it into a system. The following points are essential for this: installation space, hydraulic connection, and fluid conditioning. To enable the integration of the sensor into smaller hydraulic components such as pumps, cylinders, filter housing, etc., the sensor must not only be very small, but must be able to be connected directly to the pressure line. For this purpose, certain variants are offered, providing an integrated flow regulator, which is fixed to the flow required by the sensor. The miniaturized internal structure enables a housing shape that is common for cartridge valves in accordance with the ISO 7789 standard. The new design of the sensor allows abandoning the elastomeric material of the inner seals. Thus no special sensor variants are required for mineral oil based fluids and phosphate esters. The costs associated with logistics and warehousing are thereby reduced.



29.06.2014 - 18:18:42

Sensor 1 SN: A3560005 FW: 0.9 □ ISO 4406

ISO 20/18/14

Temperature: 30 °C 85 °F
U-Bat: 23.4 V
Free gas: 20 ppm @p
Operating Hours: 0.50 h

Sensor 2 SN:

ISO --/--/--

Temperature: - °C - °F
U-Bat: - V
Free gas: - ppm @p
Operating Hours: - h

Figure 3: Typical distribution of solid particles and gas bubbles taken with the PC application iConS (integrated contamination system)

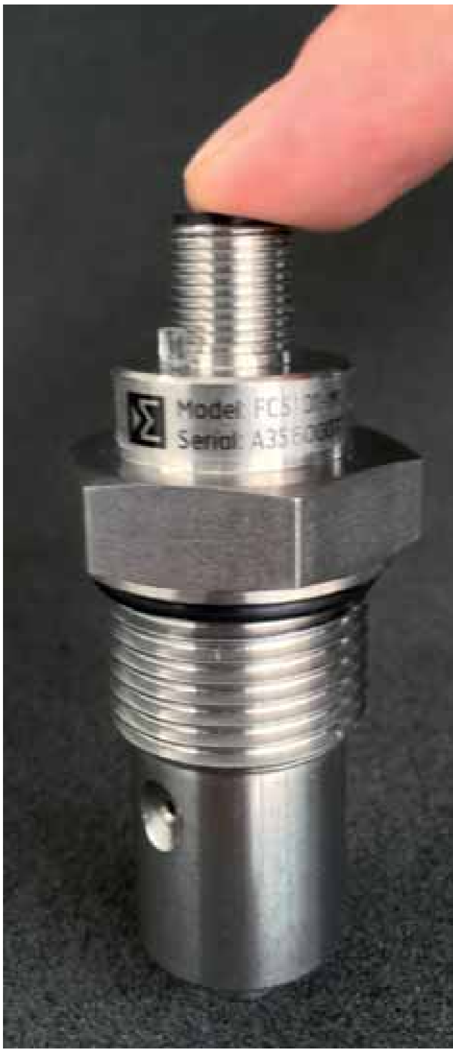


Figure 4: The world's tiniest particle counter

Electrical interfaces

The sensors of the FCS100 series come with many integrated standard interfaces: LIN, NMOS switching output, analog current and voltage output, and the fieldbus interfaces EIA-485 and CAN with connector pin-out according to CiA 303-1. The respective interface is set via a firmware update and can thus be adapted to the respective requirements. An extensive inventory of appropriate sensor models becomes therefore unnecessary. The electrical connection is made with a standard 5-pin M12 Sensor connector. The supply voltage range of 9 V to 60 V covers all battery voltages used in mobile hydraulics (12 V, 24 V, and 42 V). The total power consumption is only 500 mW.

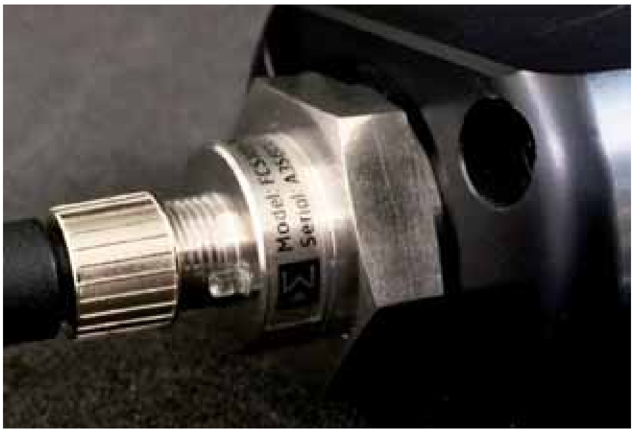


Figure 5: The sensor vanishes almost completely after installation into an appropriate cavity

Protocols for data transmission

To display all sensor data in real time, PC evaluation software is available for free. In this case the data transmission via EIA-485 uses a proprietary protocol. Data is transmitted from the sensor and immediately displayed graphically. All data is calculated in real time on the DSP of the sensor and is available without additional programming when integrated in a user's system. Also available is a galvanically isolated USB data cable that also powers the sensor. A separate power supply is not required.

For field applications there is a focus on the implementation of the CANopen protocol according to the CiA recommendations. All data will be accessible by the user. The host system will be supported by an EDS (electronic data sheet).

The low response time of the sensor to sudden changes of the state of the working fluid is possible thanks to the low dead volume of the sensor from the entrance to the measuring point of only 35 µl and the fast data collection and analysis by the DSP. This makes the sensor the first choice for bottle sampling where only a restricted volume of the fluid is on-hand.

The end of non-traceable calibration

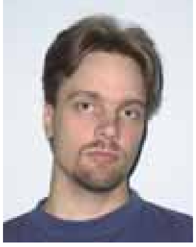
To ensure the effectiveness of quality management, it is necessary to check the used measuring instruments in fixed intervals. For example, the calibration of length measuring instruments or voltmeters is standard and is offered by all calibration services. However, this is currently not feasible for particle sensors. That is because suspensions are by their nature not traceable. The sensors have to be calibrated with an expensive reference fluid having a particular contamination. This process is also expensive and time consuming and prevents a broad application of contamination sensors.

For the FCS100 series, for the first time there will be a new system, which overcomes these drawbacks. On a glass substrate with lithographically deposited structures that are introduced into the measuring cell, the sensors can be quickly and cost-effectively calibrated by the user in the field. These microstructures are always traceable and verifiable. This allows the users to equip their systems with inexpensive contamination sensors and to ensure the proper functioning throughout the whole lifecycle. ◀

Model-based design of CANopen systems

Multiple disciplines for mechatronic system design co-exist, which hinder the utilization of software-oriented modeling principles e.g. UML. Existing modern tools may be integrated into a working tool chain.

Author



Dr. Heikki Saha

TK Engineering
P.O. Box 810
FI-65101 Vaasa
Tel.: +358-50-588-6894

Link
www.tke.fi

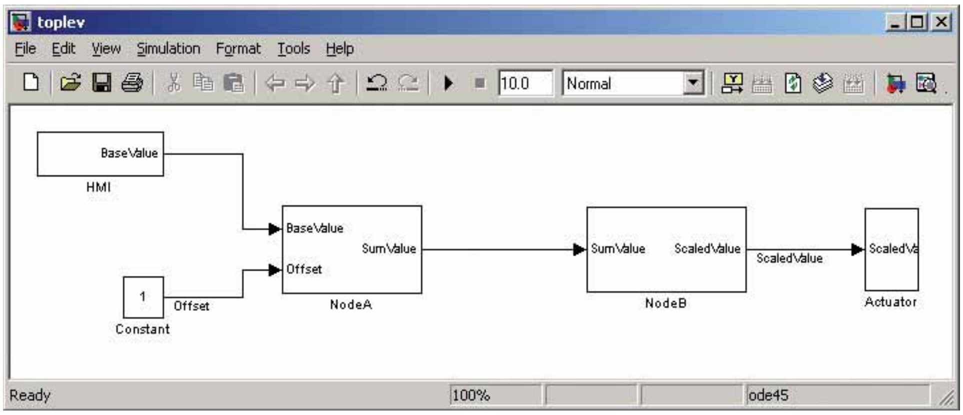


Figure 1: Example of a top-level system model consisting of two application-programmable nodes, Node A and Node B

Model-based design has become mainstream in the industry, but it has mostly been used for development of individual control functions or devices, not entire control systems. Current mechatronic systems are becoming more complex and simultaneously the requirements for quality, time-to-market, and costs have become higher. An increasing number of systems is distributed, but development is typically done device by device, without systematic coordination of system structures. Approaches to manage distributed systems with written documents have lead to inefficiency and inconsistent interfaces. Inconsistent interfaces have sometimes led to situations, where it was easier and faster for the designers to write a new software component instead of re-using an existing one. Another typical occurrence is that significant interface adjustments have to be performed during integration testing of a system. Based on such experiences, there is a demand for standardized and semantically well formed interfaces between multiple disciplines [16].

In typical mechatronic systems, multiple disciplines co-exist and none of them dominate. The multidisciplinary nature of design work makes it very difficult to utilize the modeling principles dedicated for software-oriented development, such as UML or SysML [1]. It has also been found that it is impossible to create a single tool, which is optimal for all disciplines. Instead, existing state-of-the art tools can be integrated into a well working tool chain.

The traditional way

In a typical distributed system, one function may be divided into several devices and one device may serve multiple functions. Node-centric development might be difficult because the functional distribution is not exactly known prior to development. Application-centric development and simulation provides limited efficiency because of limited testing capabilities [11]. Software-centric development without a thorough system level management will lead to serious interface inconsistencies. The old approach to managing communication interfaces is to embed communication descriptions into the application software [5]. Historically, this works with very small systems, where there is only one instance of each type of device. When devices exist more than once in a system, such an approach often leads to poor re-use of design artifacts or adoption of configuration management processes.

Model-based designs have become attractive because of the inefficiencies of the existing approaches. Though the requirement management in traditional software development has been document-centric, it has not been unusual that the requirements for the next version were collected from the source code of a ▷

previous version [18]. It has also been documented that model-based designs can reduce number defects and wasted efforts produced by current approaches.

A separate design of logical and physical structures causes challenges in managing the two parallel models and their connections without inconsistencies and still allowing incomplete models [1]. In addition, if a model-based conceptual design was used, models can be manually converted into code or control applications can be developed and tested separately, independent of each other. The main motivation for more systematic developments can be found in the assembly and service process, rather than in development, because of their higher significance [3]. Systematic configuration management enables solving serious problems e.g. during system assembly and service [3]. Systematic configuration management is required throughout the development process [18].

Existing modeling approaches

Increasing complexity of the systems requires increasing systematics during development [10]. Most defects found during the last phases of the traditional processes are caused by failures in the requirement acquisition in the early phase of the processes [10] [18]. The validation of specifications to models and model-to-code matching is easier with simulation models [9] and the use of automatic code generation with proven tools makes it possible to automate code verification and move the focus of reviews from code to models. Automatic code generation from simulation models improves the development of especially high-integrity systems [9], [10], [11]. The simulation model is actually an executable specification, from which certain documents can be generated [9], [10], [15], [18]. Higher integrity with lower effort can be achieved by validating the basic blocks and maximizing the re-use of them [15]. Conformance to corresponding standards helps to achieve required quality [15]. Simulation models can also document interfaces between structural blocks, improving consistency and enabling parallel and co-development, improving the overall efficiency [10], [12], [18].

It has been recognized that old processes produce old results [18]. New development approaches, such as a model-based design, improve the design. To achieve maximum improvements, new processes and tools are often needed. A new process with an existing, constrained design does not show benefits, but with new and more complex designs benefits can be found. A phase-by-phase approach is required to provide a learning curve. It is also important to be able to keep existing code compatible with the new code generated from models. Design re-use is one of the main things that improve productivity. The systematic management of both interfaces and behavior is mandatory in safety relevant system designs [7]. Instead of using model-based tools as a separate overlay for the existing processes and tools, automated interfaces need to be implemented between tools [18]. Connecting model-based tools with the existing legacy tools may require changes beyond built-in capabilities of the tools, increasing the effort required to maintain, develop and upgrade the tool chain.

Scope

The Simulink tool was used in the project because it is the de-facto modeling tool in research and industry

and it has open interfaces. Furthermore, it solves most of the problems found in other modeling languages and approaches [1]. One of the most significant benefits is the support of dynamic simulations. Unlike e.g. executable UML, Simulink models can be used for modeling other disciplines than software. The models can be made very simple and based on behavior only. The physical structure can be included into the model by adjusting the hierarchy of the logical model. Later on, the models can be developed to cover improved dynamics too, if required.

Because of the increasing time-to-market and functional safety requirements in machinery automation applications, higher productivity and support for model verification and re-use of designs were significant reasons for using Simulink. Such features include e.g. linking to the requirement management, model analysis, support for continuous simulation during the design process, testing coverage analysis, and approved code generation capabilities [17]. The use of Simulink models enables efficient re-use of the models for various purposes.

The main reason for using IEC 61131-3 programming languages for the evaluation is that they are well standardized, widely used in the industry, and their use has continuously been spreading. Their use in especially safety critical implementations is increasing because some of the IEC 61131-3 languages, which are considered as limited variability languages (LVL), are recommended by functional safety standards [7]. A standardized XML based code import and export format has been published recently, improving systematic design processes further.

Basically the presented approach is technology independent. CANopen was selected as an example integration framework, because the CANopen standard family covers system management processes and information storage. It is well supported by numerous commercial tool chains, which can be seamlessly integrated. The management process fulfills the requirements set for design of safety relevant control systems [7]. It is also well defined how CANopen interfaces appear in IEC 61131-3 programmable devices [2]. A managed process is required to reach the functional safety targets [7]. There is also a wide selection of various type of off-the-shelf devices on the market, enabling efficient industrial manufacturing and maintenance. Especially device profiles help re-using common functions instead of developing them again and again. In addition to the design and communication services, CANopen offers extensive benefits in the assembly and service when compared to other integration frameworks.

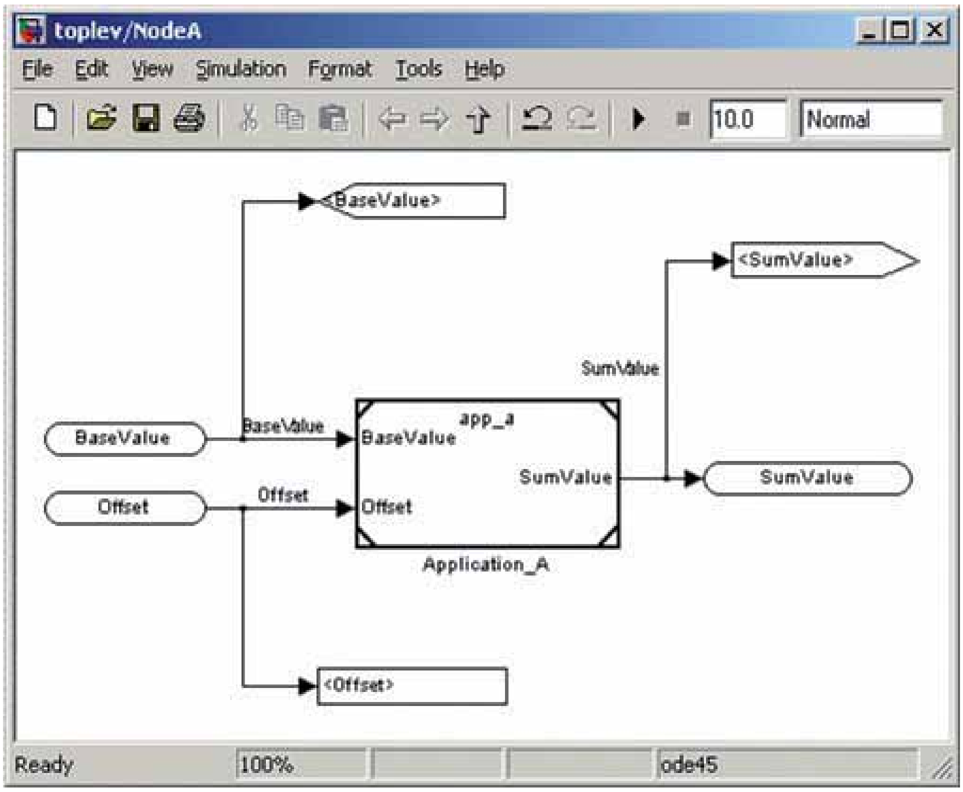


Figure 2: Example sub-model for Node A with linked application sub-model and integration interface descriptions

In this article, relevant CANopen issues are reviewed first to enable readers to understand the process consuming the presented communication description. Next, the basic modeling principles are shown. After presenting the modeling principles, the communication interface description in the model and exporting of both application interfaces and behavior are presented. Modeling details are not within the scope of this article.

CANopen issues relevant to modeling

The CANopen system management process defines the interface management through the system's life cycle from application interface description to spare part configuration download. The first task in the process is to define application software parameters and signal interfaces as one or more profile databases (CPD) [4]. Next, node interfaces defined as electronic datasheet (EDS) files can be composed of the defined profile databases. The EDS files are used as templates for device configuration files (DCF), which are system position specific and define the complete device configurations in a system. DCF files can be directly used in assembly and service as device configuration storage [19]. In addition to the DCF files, system design tools produce a communication description as a de-facto communication database format, which can be directly used in device or system analysis. A process with clearly distinguishable phases improves the resulting quality because a limited number of issues need to be covered in each step of the process [11].

Signals and parameters need to be handled differently [4] because of their different nature [14]. Signals are periodically updated and routed between network and applications through the process image [2], [4]. The process image contains dedicated object ranges for variables ▷

supporting both directions and the most common data types. The same information can be accessed as different data types. Signals are typically connected to global variables as absolute IEC addresses [2]. Signal declarations include metadata and connection information used for consumer side plausibility and validity monitoring. For parameters, metadata is used for both plausibility checking and access path declaration. All information relevant to the application development is automatically exported from the CANopen project to the software project of each application programmable device. Additionally, monitoring, troubleshooting, and rapid control prototyping (RCP) can be supported by the exported communication description. The completed CANopen project automatically serves the device configuration in assembly and service.

The process image located in the object dictionary serves also for communication between the functions or applications inside the same device [8]. It can also be shared by different field buses [6]. Software layers above the process image are not necessarily required with CANopen. The internal object access type can be defined as RWx to enable bidirectional access inside the producer device. The external access type should be defined as RWR to enable information distribution to the network. Access type RWW should always be used for incoming signals, which can be shared by multiple applications.

Parameters are stationary variables controlling the behavior of a software, their values are changed sporadically and in CANopen systems typically stored locally in each device [2], [4], [14]. Parameters of application programmable CANopen devices must always be located in a manufacturer specific area of the object dictionary. The only exception occurs if device profile compliant behavior is included. Then parameters must be located according to the corresponding device profile. It is recommended to organize application specific parameters as groups separated from the platform specific objects. Standards do not define the organization of parameter objects. Some different approaches to access parameters exist, e.g. linking global variables to objects or using access functions or function blocks.

System-level modeling in Simulink

A system model typically consists of models of a whole signal command chain, system or subsystem. The model may also contain sub-models describing behavior of e.g. hydraulics and mechanics, enabling multidisciplinary design and simulation. The main benefit of model-based design is that errors are typically found earlier than in traditional approaches [10]. Models are executable specifications enabling continuous testing [12]. When whole command chains, systems or subsystems can be tested, more practical test scenarios can be used to reveal the problems more typically found with integration tests.

The multi-disciplinary system model can also be used for initial tuning of control behavior if dynamic behavior of e.g. hydraulics and mechanics is included. After finalizing the design and initial tuning the control behavior of each device can be automatically exported into executable programs to the final HW. Because of the ease of use and automated transformations, incremental modeling and development become

▷

efficient. A simple system model is shown in Figure 1.

Node model in a system contains CANopen mapping and a referenced application behavior submodel, as depicted in Figure 2. In early stages of development, parameter and signal descriptors are not required – they do not affect on behavior, but just tag the signal or parameter to be published. Signal names and data types are directly taken from the model to the descriptions. It is presented by literature, that simulation models are commonly used for documentation and communication of interface descriptions [10]. It is important to systematically define the interfaces, because the control functions communicate through the interfaces and any inconsistency can introduce more severe global consequences that an erroneous internal behavior.

It is mentioned in the literature, that configuration management is required for simulation models [18]. One approach to arrange a well documented and proven configuration management is to implement generic simulation models and publish the all configuration parameters. The proposed approach enables the utilization of configuration management features provided by system integration framework. If CANopen is used, various model configurations can be stored as profile databases, where parameter values can be imported to the new models. Potential conflicts can be detected and solved outside the model, in the corresponding design tools.

The main benefits of the referenced models are, that they are faster in simulation [13], they enable parallel development of sub-models and can directly be used from other top-level models [12], e.g. in rapid control prototyping (RCP). RCP can significantly speed up development,

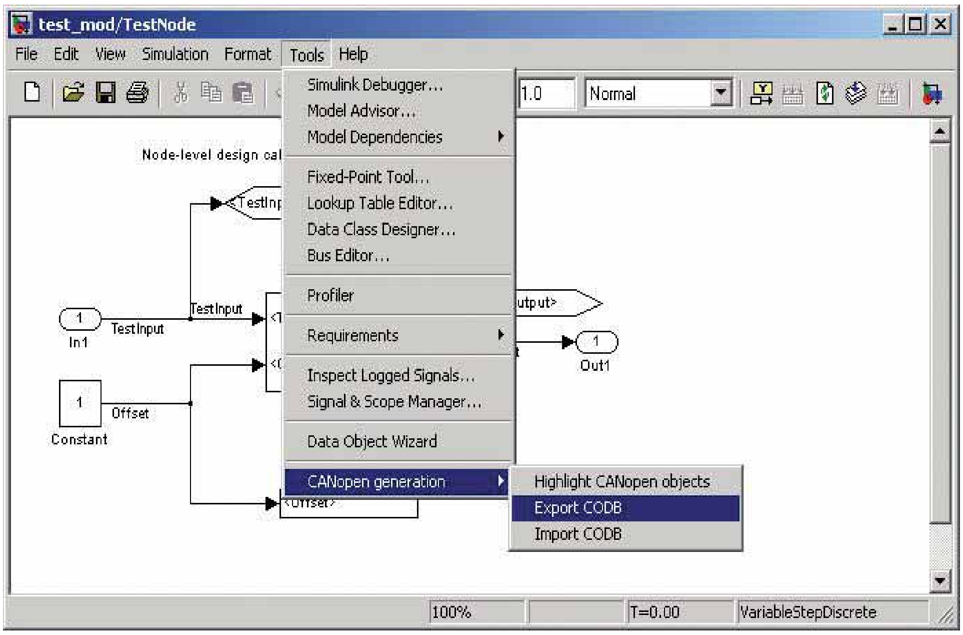


Figure 3: Example of exporting interface for Application A

because final processing performance, memory and I/O constraints do not apply [11]. Model referencing can as well be used as a reuse method of the application behavior in other models.

Preparing for export

Code generation from simulation models is a proven technology. The management of system level interfaces has not been included until now. After completing the application behavior, signals and parameters need to be defined. A dedicated blockset for such purposes has been developed. The blocks shown in Figure 2 are only markers, which are invisible to the code generation. The simulation model is independent of the integration framework and therefore only application interface descriptions are exported to framework specific tools. Such an approach enables the full utilization of the framework specific tool chain for integrating the application specific descriptions with hardware and software platform specific interface descriptions.

Signals and parameters behave differently and need to be managed accordingly [14]. Due to a thoroughly defined process image, signals may be automatically assigned into the object dictionary, but most devices have default PDO-mapping affecting the organization of the signals. Therefore it was the safest option at first to provide a manual override for automatic object assignment for the signals and parameters. The access type of signals is fixed by using direction specific blocks and the object type need not to be defined for the process image. Signals can also be introduced into e.g. device profile specific objects when standard behavior is developed. In this part of the process, compatibility with existing PLCs is as important as CANopen conformance.

Parameter management has even more deviations among different implementations. Therefore it should be possible to select the main attributes manually. The manual assignment enables parameter grouping into records and arrays, if grouping is required by applications. Access type and retain attributes are available only for parameters and their values are related to the parameter's purpose. If a parameter is intended to indicate a status, it needs to be read-only and not for retain. If a parameter's purpose is to adapt the behavior of a function, read-write access and retain are needed. Some parameters, such as output forces, require read-write access. Retain storage should not be supported, because forces should be cleared during restart for safety reasons.

Automatics can be implemented later e.g. by using target file describing object assignment rules specific to a target hardware. Development of interface standards and exchange formats will help the further development [2]. During the time of writing there are too much variations – especially in the management of parameter objects – to be covered by automatic assignment without potential need for further editing.

Minimum, maximum, and default values can be assigned for each object. They are important to be defined, because they can be efficiently re-used during further steps of the process. Those values can be given either as plain values or as variables in the Matlab workspace. Such an approach enables sharing the same metadata with ▷

application function blocks as constants linked to the same variables, but may add to the complexity of the model [14]. To speed up the modeling, value fields can be left empty, when default values are automatically used. Minimum and maximum possible values according to the object's data type are used as minimum and maximum values by default. If a default value is not defined, zero is used.

Generating exports

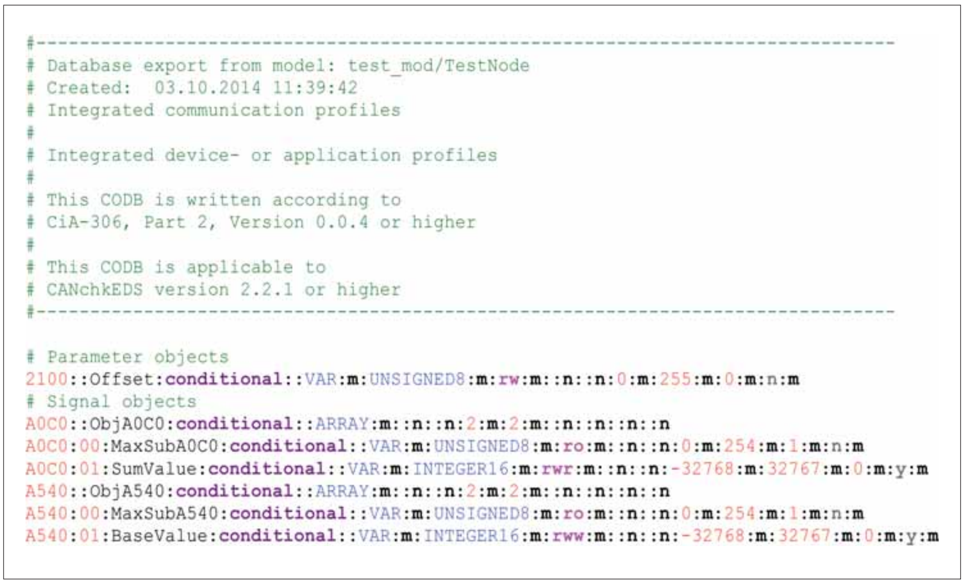
The generated application behavior needs to be isolated in a separate sub-model. Source code cannot be generated directly from the root of the referenced model. The structure of the generated code strongly depends on the internal structure of the source model [12]. The IEC 61131-3 code generation results in a single function block, where the behavior of the selected block is included. Depending on the model structure, other functions and function blocks may also be generated.

A completely fixed interface is mentioned in a case example presenting the application development improved by using fully automated code generation [11]. The more generic approach expects the management of the interfaces from the model [10] [12]. However, only application specific interfaces can be managed in the model and both hardware and software platform specific interfaces need to be managed according to the management process of selected integration framework. Applications can be developed as separate models and mapped onto the same physical node as part of the system design process. The level of modularity can be selected according to the application field. The configuration management [12] of the applications in the presented approach is supported by published application interface descriptions. The configuration management of the target system is done in a CANopen process supporting it better on the system level [11]. Calling of the application interface export of Application A is presented in Figure 3.

The resulting application parameter and signal object descriptions are shown in Figure 4. The file format in the example is a CANopen profile database (CPD) because CANopen was selected as an example system integration framework. Application interface descriptions are combined with descriptions of other optional applications, which will be integrated into the same device and the communication interface of the target device [4]. The resulting EDS-file can be used in system design as a template defining the communication capabilities of the device. System structure specific communication parameters are assigned during the system design process [2], [4].

Software integration

The first requirement is that all tools must be compatible with each other [11]. Based on experience, using standard interfaces is the easiest method to achieve a sufficient level of compatibility. Second, thoroughly defined interfaces are needed in co-development projects to get them working completely [12]. It cannot be assumed that all development is performed within a single company or department and with a uniform methodology. Third, outputs must integrate manually written, existing codes to enable either a smooth transition into model-based development or a flexible use of automatically generated and manually written code [11]. Fourth, although CANopen is currently the best integration framework in machin-



```
#------------------------------------------------------------------------------
# Database export from model: test_mod/TestNode
# Created:  03.10.2014 11:39:42
# Integrated communication profiles
#
# Integrated device- or application profiles
#
# This CODB is written according to
# CiA-306, Part 2, Version 0.0.4 or higher
#
# This CODB is applicable to
# CANchkEDS version 2.2.1 or higher
#------------------------------------------------------------------------------

# Parameter objects
2100::Offset:conditional::VAR:m:UNSIGNED8:m:rw:m::n::n:0:m:255:m:0:m:n:m
# Signal objects
A0C0::ObjA0C0:conditional::ARRAY:m::n::n:2:m:2:m::n::n::n::n
A0C0:00:MaxSubA0C0:conditional::VAR:m:UNSIGNED8:m:ro:m::n::n:0:m:254:m:1:m:n:m
A0C0:01:SumValue:conditional::VAR:m:INTEGER16:m:rwr:m::n::n:-32768:m:32767:m:0:m:y:m
A540::ObjA540:conditional::ARRAY:m::n::n:2:m:2:m::n::n::n::n
A540:00:MaxSubA540:conditional::VAR:m:UNSIGNED8:m:ro:m::n::n:0:m:254:m:1:m:n:m
A540:01:BaseValue:conditional::VAR:m:INTEGER16:m:rww:m::n::n:-32768:m:32767:m:0:m:y:m
```

Figure 4: Interface descriptions for parameters and signals of Application A as a CANopen profile database

ery applications, upgrade paths and additional supported integration frameworks should also be possible.

A generic approach does not support predefined signaling abstraction used in some implementations [11]. Instead, application specific abstractions need to be generated from the model and developed further in the CANopen process, where physical platform specific and communication specific details can be integrated most efficiently into a complete description of a device's communication interface. That includes necessary information from the rest of the system [4] [20]. Finally the communication abstraction is imported as an IEC code into a development tool. Manual coding is required only for connecting the exported application behavior into communication and I/O abstraction layers. The approach follows a standardized process enabling integration of commonly used tools, which is also recommended in the relevant literature [18]. Relying on a standardized process enables a simple adaptation natively supported by the tools and heavy tool customizations are avoided, which confirms the findings already presented in the relevant literature [18].

The remaining manual integration work is minimal, mainly consisting of connecting application signals and parameters to the communication abstraction layer. Moreover, signal and parameter metadata – minimum, maximum, default values, and signal validity – if used by application behavior, also need to be connected manually to the relevant application function blocks. Fixed connections are not performed, because such information is not necessarily required for all signals and parameters. Including complete metadata for all signals and parameters with plausibility checking may require too much memory and processing power. An automatic connection would also violate the requirements of flexible mixing of manually written and automatically generated code [18].

Discussion

An approach to including public interface descriptions into the same model with system behavior divided into multiple application has been presented. Such an approach enables an efficient system level interface management, which serves the design process by enabling the export of application specific signal and parameter descriptions. Furthermore, the be- ▷

References

1] Laakso M., Distributed System Design Flow: Fieldbus Modeling, Master's thesis, TUT, 2008, 78 p.

[2] Saha H., Improving development efficiency and quality of distributed IEC 61131-3 applications with CANopen system design, Proceedings of 13th iCC, CiA, 2012, pp. 10-15 – 10-21

[3] Saha H., Benefits of intelligent sensors and actuators throughout the systems life cycle, The Twelfth Scandinavian International Conference on Fluid Power, May 18-20, 2011, Tampere, Finland, ISBN-978-952-15-2517-9, pp. 169 – 181

[4] Saha H., Wikman M., Nylund P., CANopen network design and IEC 61131-3 software design, CAN-Newsletter 3/2009, CiA, 2009, pp. 52 – 58

[5] Tisserant E., Bessard L., Trelat G., Automated CANopen PDO Mapping of IEC 61131-3 Directly Represented Variables, Proceedings of 12th iCC, CiA, 2008, pp. 06-08 – 06-13

[6] Rostan M., Hoppe G., Generic Fieldbus Application Program Interface for Windows, Proceedings of the 7th iCC, CiA, 2000, 7 p.

[7] Safety of machinery. Functional safety of safety-related electrical, electronic and programmable electronic control systems, EN 62061, 198 p.

[8] Additional application layer functions, Part 4: Network variables and process image, CiA-302-4, CiA

[9] Conrad M., Verification and Validation According to ISO 26262: A Workflow to Facilitate the Development of High-Integrity Software, SAE,

[10] Murphy B., Wakefield A., Friedman J., Best Practices for Verificzation, Validation, and Test in Model-Based Design, SAE, 2008-01-1469

[11] Thate J. M., Kendrick L. E:, Nadarajah S., Caterpillar Automatic Code Generation, SAE World Congress, 2004-01-0894

[12] Anthony M., Friedman J., Model-Based Design for Large Safety-Critical Systems: A Discussion Regarding Model Architecture

[13] Nadarajah S., Large Scale Modeling and Simulation of Propulsion Systems, SAE, 2007-01-1645

[14] Anthony M., Behr M., Model-Based Design for Large High Integrity Systems: A Discussion on Data Modeling and Management, AAS 10-023

[15] Anthony M., Behr M., Jardin M., Ruff R., Model-Based Design for Large High-Integrity Systems: A Discussion on Verification and Validation

[16] Markkula M., Rokala M., Palonen T., Alarotu V., Helminen M., Koskinen K. T., Utilization of the Hydraulic Engineering Design Information for Semi-Automatic Simulation Model Generation, Proceedings of The 12th Scandinavian International Conference on Fluid Power, 2011, ISBN 978-952-15-2522-3

[17] Erkkinen T., Conrad M., Safety-Critical Development Using Automatic Production Code Generation, SAE 207-01-1493

[18] Dillaber E., Kendrick L., Jin W., Reddy V., Pragmatic Strategies for Adopting Model-Based Design for Embedded Applications, SAE 2010-01-0935

[19] Saha H., Accelerated transfers of CANopen projects into assembly and service, CAN Newsletter 4/2012, CiA, 2012, pp. 17-20

[20] Saha H., Experimental CANopen EEC management, CAN Newsletter 1/2013, CiA, 2013, pp. 12-18

havior of each application can be generated from the same model. Application programs with communication abstraction layers can be developed simply by combining interface descriptions and application code modules. The uniform and automated management of system integration interfaces improves the development process and enables a model-based design of entire systems instead of a design of individual applications. In addition to behavioral errors, information interchange inconsistencies can be found earlier, which reduces failure costs. Moreover, higher system-wide safety integrity can be reached through the presented approach more comprehensively than before.

The use of proven tools and standardized file formats enables an efficient re-use of design information throughout the design process. Small changes during the process are inherently made directly into the CANopen project – DCF-files. Changes can be updated backwards to the corresponding EDS-file easily with existing tools. Updated EDS-files enable node re-use of the devices with the most recent changes [4]. Application interfaces defined as CPD files can be updated by extracting the defined part of an EDS-file into the corresponding CPD, which enables application level re-use. The changes can be read back from CPD into a simulation model. The signal or parameter name and data type introduce a problem, because in export they are taken from the model. However, if additional changes are allowed, incomplete back annotations from CPD into the simulation model can be performed. The problem is not significant, because the model should be the master version for both behavior and interfaces anyway [10].

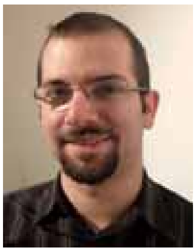
Model-based development and model referencing enables the direct re-use of application behavior as referenced models for other purposes, such as RCP and education simulators. Source code generated from the model can also be re-used indirectly in code modules. Code generation supports several programming and hardware description languages, which also enable the optimization of partitioning between hardware and software implementations.

Although systematic, system-wide signal and parameter management as an integral part of model-based designs has been implemented, further development is needed. From a process efficiency point of view, it is most important to develop the automatic assignment of parameter object indexes. Such a development should be tightly coupled with the integration framework specific standardization work. Such improvements, like an automatic connection of applications into communication and plausibility checking of signals using partial value range, will be implemented in the future. Including I/O abstractions is also an interesting topic for the future. It is also possible to add support for other system integration frameworks than CANopen. Based on current knowledge, a fully automatic software development requires such tight constraints for hardware and software components that such a development is not important.

CAN driver for Windows with analyzer capability

Thread-safe virtualization, the access to .NET, and the built-in analyzing tool of a CAN driver for C/C++ enable an object-oriented software development of CAN applications.

Authors



Martin Andermann

Hartmut Keller

F&S Elektronik Systeme GmbH
Untere Waldplätze 23
DE-70569 Stuttgart
Tel.: +49-711-123-722-0
Fax: +49-711-123-722-99
info@fs-net.de

Link
www.fs-net.de

Despite the fact that CAN has its origins in the automotive sector, it has found its way to the embedded market. This is especially the case when sensors have to be queried or devices have to be controlled and if focus is on high transmission security. Today CAN is also used in medical devices or automation technology. F&S Elektronik Systeme is a manufacturer of embedded systems in the form of single board computers (SBC) and system-on-modules (SOM). These provide at least one CAN interface and are available with Linux, as well as Windows Embedded Compact operating systems.

System-on-chip manufacturers offer CAN drivers for Windows, but usually only with support of their own controllers, resulting in drivers that differ from each other. F&S has developed a CAN driver for C/C++, which offers the same interface on all manufacturer's boards, regardless of the used CAN controller. They also offer a suitable class library for .NET, simplifying the use of the driver with e.g. C#. The CAN driver provides an object-oriented interface to the driver, including CAN interface construction and destruction, support for exceptions in case of errors and taking advantage of overloaded functions.

The driver uses the concept of virtual send and receive channels. These channels give all applications and threads their own access to the CAN network. All channels are independent from each other. Each channel has its own transmit and receive buffer, as well as a separate acceptance filter. For example, on multi-core CPUs (central processing unit) the efficiency of the message processing can be increased by using multiple parallel threads. Each thread opens its own file handle, provides its own acceptance filter, and then receives its own messages.

Several distinct programs can access the CAN controller, but since a CAN controller usually does not receive its self-sent messages, a message sent on the network by one of these programs cannot be seen by other ▷

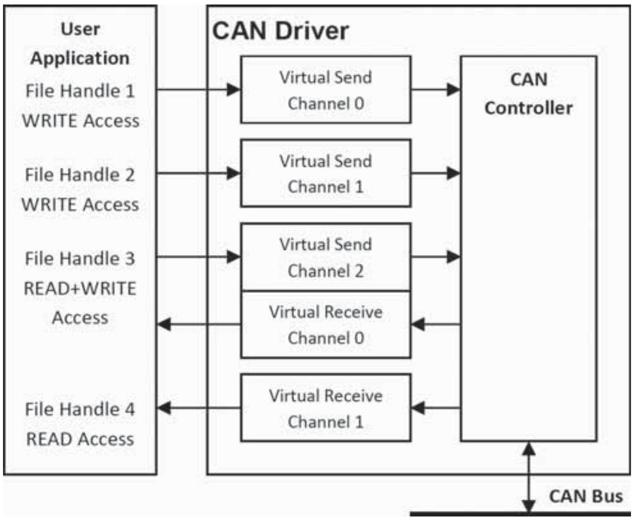

User Application
File Handle 1 WRITE Access
File Handle 2 WRITE Access
File Handle 3 READ+WRITE Access
File Handle 4 READ Access
CAN Driver
Virtual Send Channel 0
Virtual Send Channel 1
Virtual Send Channel 2
Virtual Receive Channel 0
Virtual Receive Channel 1
CAN Controller
CAN Bus

Figure 1: CAN driver with virtual channels for read and write access

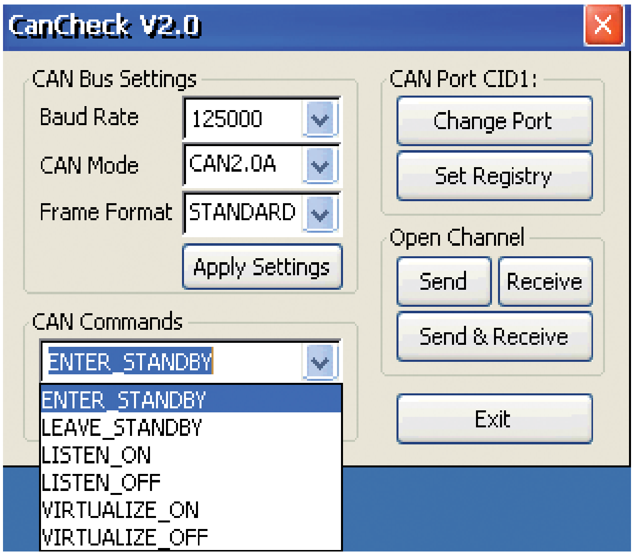
local programs. To overcome this disadvantage, the driver can be set into a so-called Virtualize mode. Now the messages are not only sent out to the physical CAN network, but are also forwarded by the driver to the receive channels of all the other local programs. From the programs' point of view, each of them now has its own (virtual) CAN controller. The Virtualize mode can be (de-)activated during the runtime. The CAN driver also offers the possibility to switch to a Listen Only mode. In this case, the transmitter of the controller gets deactivated. This prevents an accidental sending of messages.

The Enter Standby command puts the CAN controller into a sleep mode to reduce electrical consumption. This mode is left automatically, when there is communication on the network, or manually, with the Leave Standby command. All these mode changes are reported as events to all virtual receive channels.

To find out whether there is a communication malfunction on the CAN network or an error in the own software, a CAN analyzer, in form of external hardware, is often required. The built-in CanCheck tool in the driver makes such an analyzer unnecessary in many cases. It can either be set as a communication partner on a second board, or directly on the development board. The tool can be used like a sniffer program, showing the live traffic on the CAN network. The received data can be saved in a log file for later analysis. The tool can also generate arbitrary CAN messages for test purposes. It is a graphic tool, which is operated by mouse or touch. During development, the focus was put on compact dialogs, so the program is also applicable on small displays. If there is no display available, company's boards use a virtual display, which can be made visible on a PC with a remote desktop connection (e.g. CerDisp). Used in this way, the tool can be controlled remotely.

In the "CAN Bus Settings" field of the tool's main window, the user can adjust the network settings, for example the CAN port (if there is more than one controller available), the used bit-rate, or whether the frames with 11-bit or 29-bit CAN-ID (identifier) are used. The settings can be permanently saved directly from the dialog into the Windows registry. The "CAN Commands" such as Standby, Listen Only and

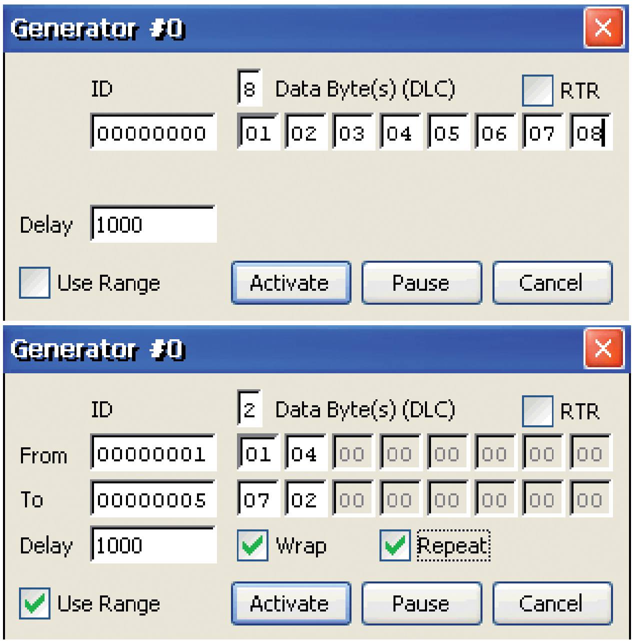
▷


CanCheck V2.0

CAN Bus Settings
Baud Rate: 125000
CAN Mode: CAN2.0A
Frame Format: STANDARD

CAN Commands: ENTER_STANDBY
ENTER_STANDBY
LEAVE_STANDBY
LISTEN_ON
LISTEN_OFF
VIRTUALIZE_ON
VIRTUALIZE_OFF

Figure 2: The main window


Generator #0

ID: 00000000
8 Data Byte(s) (DLC): 01 02 03 04 05 06 07 08
□ RTR
Delay: 1000
□ Use Range

Generator #0

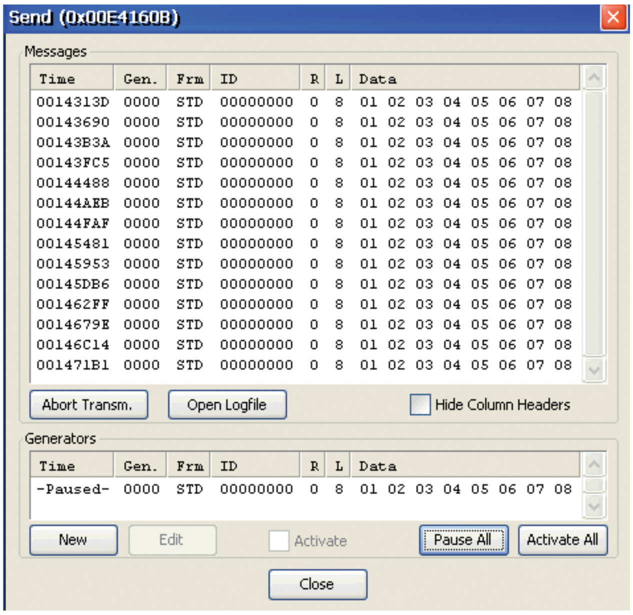
ID From: 00000001 — 2 Data Byte(s) (DLC): 01 04 00 00 00 00 00 00
To: 00000005 — 07 02 00 00 00 00 00 00
□ RTR
Delay: 1000
☑ Wrap ☑ Repeat
☑ Use Range

Figure 3: The send generator window

Virtualize can be selected and sent from a drop-down list. Every click on one of the buttons in the "Open Channel" field creates a new send channel, a receive channel, or a combined send/receive channel each in its own window.

The send functionality of the tool offers the possibility to create multiple so-called send-generators. The generators run separately and can also be paused or activated separately. The easiest case when creating a generator is to transfer a message with a constant CAN-ID and up to eight constant data bytes. By defining a start and end value, it is also possible to automatically increment and decrement CAN-ID and data bytes individually. The wrap-over function and repetitions can be activated. The "Delay" value defines the pause between the separate CAN messages of each generator. By setting the RTR (remote transmission request) flag, it is possible to send CAN request frames.

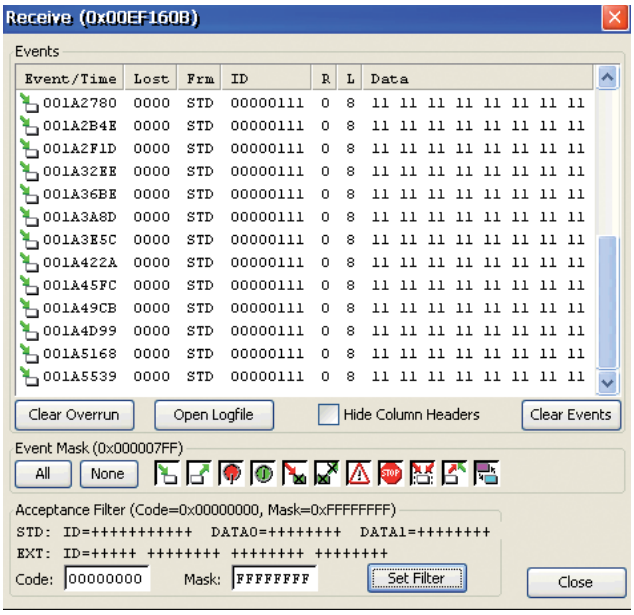
In the receive window, one can either see the complete communication on the CAN network, or messages with a certain CAN-ID (or CAN-ID range) only. This can be done by adjusting the acceptance filter. The tool decodes the acceptance filter to show which bits are actually checked (0/1) or generally accepted (+). Different CAN events like "Message received", mode adjustments, as well as errors like "Overrun" or "Arbitration lost" are visualized with different symbols. Toggling the buttons with the event symbols enables or disables the appropriate event, i.e. the event is shown in the events list or it is ignored. This functionality can be used to reduce the list to error messages only. ◀


Send (0x00E4160B)

Messages
```
Time      Gen.  Frm  ID        R  L  Data
0014313D  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
00143690  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
00143B3A  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
00143FC5  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
00144488  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
00144AEB  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
00144FAF  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
00145481  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
00145953  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
00145DB6  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
001462FF  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
0014679E  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
00146C14  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
001471B1  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
```
□ Hide Column Headers

Generators
```
Time      Gen.  Frm  ID        R  L  Data
-Paused-  0000  STD  00000000  0  8  01 02 03 04 05 06 07 08
```
□ Activate

Figure 4: The send window


Receive (0x00EF160B)

Events
```
Event/Time  Lost  Frm  ID        R  L  Data
001A2780    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A2B4E    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A2F1D    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A32EE    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A36BE    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A3A8D    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A3E5C    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A422A    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A45FC    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A49CB    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A4D99    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A5168    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
001A5539    0000  STD  00000111  0  8  11 11 11 11 11 11 11 11
```
□ Hide Column Headers

Event Mask (0x000007FF)

Acceptance Filter (Code=0x00000000, Mask=0xFFFFFFFF)
```
STD:  ID=+++++++++++  DATA0=++++++++  DATA1=++++++++
EXT:  ID=+++++ ++++++++ ++++++++ ++++++++
```
Code: 00000000 Mask: FFFFFFFF

Figure 5: The receive window

CAN FD: Improved residual error-rate

Classical CAN provides several error-detection mechanisms. They determine the residual error rate. CAN FD uses the same mechanisms and an additional one that reduces the probability of undetected errors further.

Related articles

- Florian Hartwich, Robert Bosch GmbH
  CAN with flexible data-rat
  CAN Newsletter (print), June 2012
- Magnus-Maria Hell, Infineon Technologies
  The physical layer in the CAN FD world
  CAN Newsletter (print), March 2014
- Bernd Elend, NXP
  CAN FD: Impact on system design
  CAN Newsletter (print), June 2014

References

[1] P. Koopman and T. Chakravarty: CRC polynomial for embedded networks, International Conference on Dependable Systems and Networks (DSN-2004)

[2] F. Yang: Residual error rate of CAN FD (unpublished paper), June 2014

One of the most powerful error-detection mechanisms is the CRC (Cyclic Redundancy Code) embedded in each CAN data/remote frame. The 15-bit polynomial used in Classical CAN provides a Hamming Distance (HD) of six, meaning it can detect all randomly distributed 5-bit failures in a single frame. It can also detect any 15-bit burst errors.

CRCs are a first line of defense against data corruption. The achievable HD depends on the length of the data to be protected. The chosen 15-bit CRC is capable of detecting 5-bit errors when the protected data has 112 bit or less [1]. However, CRCs protect data only if the bit string has the very same size (the same number of bits) on the transmitting and the receiving sides. In Classical CAN you find cases in which two bit-flips (generating/eliminating stuff conditions) can lead to a valid frame from the view of the CRC. The reason for this is that the dynamic stuff-bits are not considered in the CRC calculation.

To overcome this "weakness", the CRCs that are used in the CAN FD protocol consider the dynamic stuff-bits. Additionally, in the CRC field fixed stuff-bits are used. Unfortunately, considering the dynamic stuff-bits in the CRC calculation allows a situation in which a single error is not detectable. This happens for example, when a local glitch leads to a missynchronisation of a receiving node while the CRC generator registers are at "0…0". If such a glitch coincides with a stuff condition, it may happen that the receiving node reads the bit sequence "00000i" (i = stuff-bit) as "00001". In other words, this is a shortening of the frame by skipping a bit. Of course, this scenario is not likely. Nevertheless, this has a negative impact on the residual error-rate. Engineers working with Renesas found such scenarios: They showed that a corruption of the Start-of-Frame bit is not detectable by means of the CRC mechanism. Subsequently, experts from Bosch showed that this may also happen at other positions of the frame.

"To solve this weakness of the CAN FD protocol, we proposed to introduce a stuff-bit counter (SBC)," said Dr. Arthur Mutter from Bosch. "The receiving node needs to know the total number of transmitted bits for each frame. From the protocol specification and the DLC (data length code) the receiver knows the length except for the number of dynamic stuff-bits. It is sufficient to transmit the stuff-bit count modulo 8, because an HD of "just" six is required. The three SBC bits are able to detect up to seven lengthening or shortening errors, which otherwise could remain undetected, if they coincided with stuff conditions."

The SBC bits belong to the CRC field where fixed stuff-bits are used. They are transmitted before the CRC bits. The SBC bits are not part of the Classical CAN stuff-bit rule, because a stuff-bit in the SBC cannot be included in the counting. The SBC bits are protected by the CRC calculation.

Safeguarding of the SBC

When a stuff-bit is dropped or inserted by synchronization failure, the CRC is corrupted. If in the same frame a bit-flip falsified the stuff-bit count, the receiver may not be able to detect this error. This is why the SBC needs to be safeguarded. There are two safeguards implemented now:

- Adding an even parity–bit
- Gray coding the stuff-bit count

▷

Table 1: The stuff-bit counter and its parity bit is located in the CRC field in front of the CRC polynomial, which starts with an fixed stuff-bit

Stuff-bit count (modulo 8)	First bits of the CRC field		
	SBC value	SBC parity bit	Fixed stuff-bit
0	000	0	1
1	001	1	0
2	011	0	1
3	010	1	0
4	110	0	1
5	111	1	0
6	101	0	1
7	100	1	0

The table shows the SBC bits, the parity-bit, and the following fixed stuff-bit. "The parity check and the fixed stuff-bit (always with the inverted value of the preceding bit) detect any single-bit error of these bits," explained Dr. Mutter. "The same holds for two bit-flips, if at least one of the bit-errors occurs in the parity-bit or the following fixed stuff-bit. If two bits in the Gray-coded SBC are corrupted, this results in a stuff-bit count value with a difference of at least 2. This is detected through a comparison with the internally counted stuff-bit value. The minimum number of bit-errors that could remain undetected is four. This happens only if two bit-flips in the Gray-coded SBC coincide with two stuff-bits dropped or inserted."

The receiver checks the received stuff-bit count (modulo 8) with its internal count and also performs a parity check. A mismatch during the SBC comparison is treated the same way as a detected CRC error. This means that the related Error flag is transmitted after the ACK field.

Detection of all single-bit errors

Other 2-bit errors can cause undetected faulty messages too. If the IDE (identifier extension), the FDF (FD frame), or one of the DLC bits are corrupted and in the data field one of the stuff-bits is evaluated as recessive by mistake, it is possible that a "shorter" valid frame is accepted by the receiving node. This has been described in detail by the Chinese researcher Fuyu Yang [2]. Of course, the additional bits would cause an error condition. But the "faulty" message in front has already been accepted and eventually processed (depending on the acceptance filtering settings). Even if these scenarios are very unlikely, they need to be considered when calculating the residual error-rate. In this case, the receiver checks the CRC while the transmitter sends data bits. There is a probability that the perceived CRC field matches with random data bits depending on its length. The residual error-rate of CAN FD is expected to be much lower than in Classical CAN because the CRC field is much longer in CAN FD. In Classical CAN this critical bit sequence is 15 bit long while in CAN FD it is 27 bit in a frame with CRC-17 and 32 bit in a frame with CRC-21.

In order to improve the CRC checking, the initialization vector for the CRC-17 and CRC-21 has been changed from (0..0) to (100..0), where the "1" is at the most significant bit position followed by "0". All these improvements will be introduced in the next version of ISO 11898-1 CAN data link layer standard, which is currently under review.

Holger Zeltwanger

Secure communication for CAN FD

Encrypted data transmission is not yet the norm in vehicle networks. Vector has conceived an implementation for secure communication over CAN. Protection goals were authentication and preventing replay attacks.

Author

Armin Happel

Principal Software Development Engineer
Vector Informatik GmbH
Ingersheimer Str. 24
DE-70499 Stuttgart
Tel.: +49-711-80670-0
Fax: +49-711-80670-111

Link
www.vector.com

In today's vehicle networks, data transmission is for the most part performed without any special security measures. Because of this, it is possible to read out the data transmitted in raw format or to even play it into the bus system in modified form if you have direct access to the vehicle bus. Encrypted data transmission would not only ensure that this information could only be evaluated by authorized recipients. At the very least, it would also make it much more difficult to intercept or alter the messages.

Media reports about vehicle manipulation [1], [2] raise the question of whether data in the vehicle network can actually be influenced by manipulation. Can a manipulated device or internally implanted device with a remote control function influence vehicle behavior? And what countermeasures can be taken to prevent such manipulations?

Today's vehicles are highly complex systems, which consist of networked sensors and actuators and continually transmit important data over bus systems. In the vast majority of cases, the information being transmitted is in raw data format. A plausibility check, if such a check is even possible, has limited effectiveness. The receiver is unable to verify whether the data was ▷

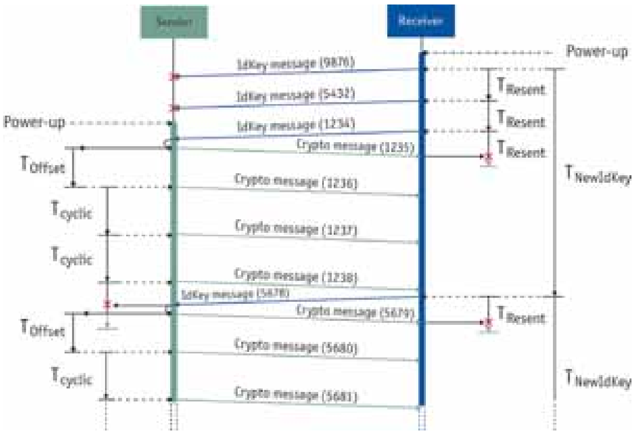


Figure 1: Message transmission and timing of encrypted communication

actually supplied by the desired sender or whether it was fed in by an outside electronic control unit, i.e. whether it is authentic data. The data is freely accessible as well, so an analysis of the bus information can be used to determine signal contents. The transmission is neither confidential nor authenticated.

This was the problem that engineers at Vector were confronted with. Their task was to come up with an implementation for secure communication over a CAN network which can be used flexibly and can also be integrated with Autosar-3.x basic software. Protection goals were authentication and preventing replay attacks. It was also desirable to implement communication that cannot be monitored

For the encryption method, the specialists chose the AES algorithm [3]. From today's perspective, this method is considered cryptographically secure. It involves symmetrical block encryption with a block length of 128 bits. It generates 16 bytes or a multiple of 16, which the sender transmits to the receiver. An additional advantage is that some microcontrollers already have very fast hardware-based implementations of this algorithm.

Since a CAN message can transmit a maximum of 8 data bytes per frame, a decision was made to utilize the ISO transport protocol (TP) that was already included in the communication stack for the transfer. To simplify the configuration and reduce protocol overhead, a unidirectional communication with a fixed 1:1 relation between sender and receiver was chosen.

Symmetrical encryption requires that both the sender and receiver have the same key. The software modules that are used permit dynamic allocation of the keys at runtime, so that the user or OEM can freely choose them. A higher-level method such as a (asymmetrical) key exchange method might be implemented, or a static allocation might be made such as in end-of-line programming. Whenever an ECU is replaced and a vehicle specific key is used, the new ECU must be set up by an authorization method, which keeps the key confidential under all circumstances.

Preventing replay attacks

In this configuration, an encrypted transmission of messages is now possible, where the information is, however, still purely static, i.e. a unique key text can be assigned to the plain text signals. This means that replay attacks, i.e. recording excerpts of a desired communication and replaying it into the system at a later time, can still be made. That is because the receiver cannot check whether the message actually originates from the sender at this point in time. To make checking possible, at the start of communication the receiver generates a random value – which is referred to as the ID key in the following – and it communicates this to the sender. The sender increments the value with each transmission and appends it to the transmit message. When the message arrives, the receiver checks whether the ID key matches the expected value. If it does, it processes the message; otherwise it rejects it. To tolerate possible message losses, the receiver will also accept a slightly higher value. This means that the counter in the transmit message continually alters the encrypted data even if the signal contents remain the same (Figure 1).

Depending on the word width of the ID key and the frequency with which the message is sent, overruns of the counter value might be expected in the ▷

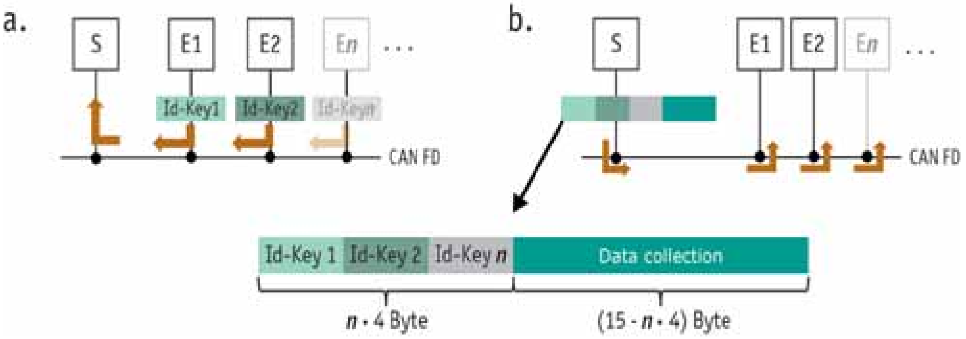


Figure 2: ID keys of multiple receivers in the use of CAN FD

message, which would lead to repeated transmission of the encrypted message. To avoid this, the ID key is only valid for a certain time period. When this period expires, the receiver must generate a new value and communicate it to the sender. Immediately after receiving a new ID key, the sender transmits the encrypted message. This means that the receiver is also able to initiate repetition of a message, such as if the received ID key does not agree with the internal key, and this reduces latency times. Although the sending node receives and considers new ID key messages for a time T(offset), to avoid an overload of the bus system such messages do not immediately lead to resending of the encrypted message. To make the protocol more robust, the receiving side uses the timer T(Resent) to monitor the response of the sender with the new counter value. If it does not get an acknowledgment message from the sender, the receiver generates a new ID key and resends it. This makes it possible to detect even a brief failure of the sending ECU and shortens the time for resending. It also avoids storage of the ID key in non-volatile memory.

Data transmission without segmentation

There is a significant disadvantage associated with segmented data transmission in CAN over the ISO-15765 transport protocol. Transmission time is increased, and this method is restricted to a fixed 1:1 relationship, because segmented data transmission over ISO-15765 is very difficult to implement with multiple nodes. CAN FD on the other hand enables simultaneous transmission of the entire encrypted message to multiple receivers [4]. Each receiver needs the same symmetrical key to decrypt the encrypted message. Two variants of the ID key for authentication come into consideration: either all receivers agree on a commonly agreed value, or all receivers independently generate and send their ID key to the sender. The sender manages all counters and appends them to the data message. The positions of the counter values within the encrypted message must be uniquely assigned to the receivers.

Figure 2 shows data transmission for multiple receivers. First, the receivers transmit their randomly generated start values to the sender. The sender then increments all ID keys for each send cycle and insert them into the encrypted

Literature

[1] http://www.chip.de/news/CAN-Hacking-Tool-Autos-hacken-fuer-20-Dollar_67066892.html [only German]

[2] http://www.can-newsletter.org/engineering/engineering-miscellaneous/140822_list-of-potentially-vulnerable-cars_blackhat/

[3] Advanced Encryption Standard (AES), FIPS PUB 197

[4] CAN with Flexible Data Rate – Specification Version 1.0, Robert Bosch, GmbH; April, 2012 http://www.bosch-semiconductors.de/en/ubk_semiconductors/safe/ip_modules/can_fd/can.html

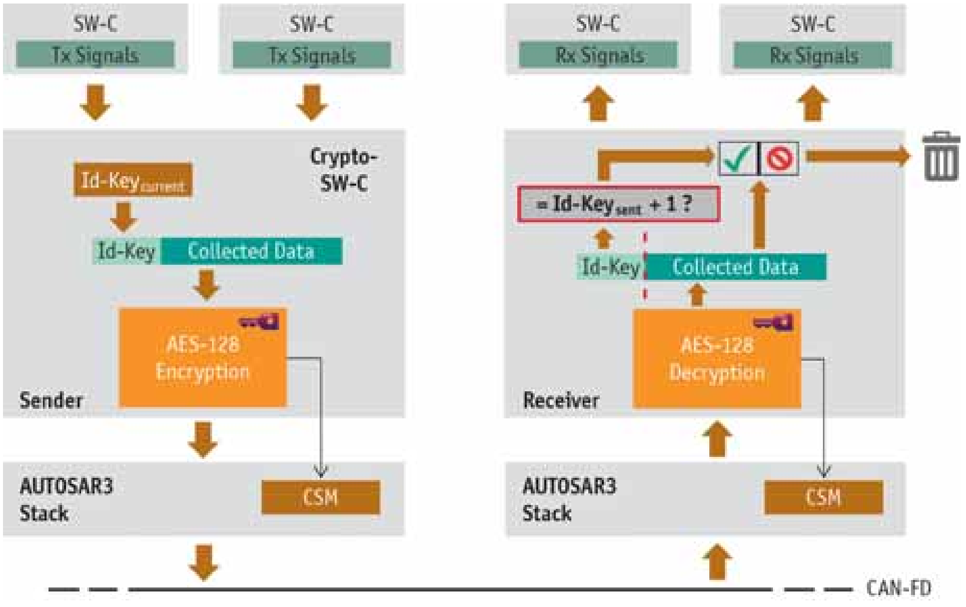


Figure 3: Software components for encrypted transmission

message at the predefined positions. The relevant receiver then checks its ID key and accepts the data or rejects it (Figure 2).

However, as the number of receivers increases, this reduces the message space that remains for useful data. The number of useful data bytes is also highly dependent on the selected word width of the ID key. The communication timing illustrated in Figure 1 was applied. It only required a modification for the sender in receiving the ID key. Instead of immediately transmitting the encrypted message, the sender waits for a configurable time T(IdKeyReply) to allow time for any other ID key messages from other receivers. The special case T(IdKeyReply)=0 covers the original method.

Vector implemented the protocol for CAN FD in a CANoe environment. The specialists subjected the protocol to extensive tests using this software tool for development, simulation, and testing of ECUs and networks. Along with the required robustness against replay attacks, another focus was to study message losses, failure, and re-entry of sender and receiver as well as timing errors and burst attacks. In all of these cases, the encryption system provided a stable transmission.

Summary and Outlook

In CAN FD, in particular, it took relatively little effort to implement robust transmission of encrypted data with multiple nodes, and this method can also fit into an existing Autosar environment. One disadvantage is the serialization and deserialization of the data on the application level (Figure 3), which means that modeling properties of the RTE cannot be used any longer for individual signals. The classic points of attack on such systems must still be kept in mind. They include, for example, weak random number generators for the ID keys (at startup) or spying the symmetrical keys.

In the security technology world, the AES-128 algorithm is considered secure for the near future, and its implementations are mature or will even be supported by hardware accelerators. The method presented here makes attacks on the CAN (FD) communication much more difficult, and manipulation is hardly possible without “insider knowledge”. It has already been in production use for several years, and it also has led to favorable classification of the relevant vehicle for insurance premiums. In this case, security not only protects data; it even offers a direct cost advantage to the end user.

In the near future, remote connections such as Car2x communication, WLAN, Bluetooth and Internet will continue to grow and will necessitate much more stringent requirements for IT security. These access modes must be made secure against attacks and must not permit any remote manipulation. This is especially true of information to driver assistance systems, which rely on reliable messages from other traffic participants and/or the infrastructure. ◄

Subscription

Please use the following classifications for filling in your subscription form:

A Position in company

0. Other position
1. Director
2. Technical manager
3. Marketing manager
4. Sales manager
5. System designer
6. Device designer
7. Purchasing manager

B. Company‘s CAN business

0. Other business
1. Semiconductor manufacturer
2. Device manufacturer
3. Software house
4. System integration
5. Service provider
6. End-user
7. Research

C. CAN application interest

1. Passenger cars
2. Heavy-duty vehicles
3. Rail vehicles
4. Maritime vehicles
5. Aircraft/Aerospace vehicles
6. Power generation systems
7. Factory automation systems
8. Process automation systems
9. Industrial machine control systems
10. Construction machine control systems
11. Embedded control systems
12. Building automation
13. Door control systems
14. Lift control systems
15. Medical devices and systems
16. Science and research systems

D. Company size

1. 1 - 9 employees
2. 10 - 49 employees
3. 50 - 99 employees
4. 100 - 499 employees
5. 500 - 999 employees
6. 1000 - 4999 employees
7. 5000 - 9999 employees
8. 10000 - 99999 employees
9. more than 100000 employees

CAN Newsletter

I hereby subscribe to the free-of-charge CAN Newsletter for the next four editions (published in March, June, September, and December of every year).

□ Print version □ PDF version

Company

Name

Address

City, ZIP

Phone (Country code - Area code - Number)

Fax (Country code - Area code - Number)

E-mail

URL

Position (see A) Application (see C)

Business (see B) Size (see D)

□ I like to receive CiA’s Weekly Telegraph

Please send your subscription form to CAN in Automation (CiA) GmbH, Kontumazgarten 3, DE-90429 Nuremberg, Germany, or fax it to +49-911-928819-79 or e-mail it to headquarters@can-cia.org. You may also subscribe online at www.can-cia.org.

PROEMION
Telematics Systems

CANlink® mobile with REAL-TIME ¬

Redefined fleet management and diagnostics

REAL-TIME / DASHBOARD

> Real time monitoring of CAN data
> Worldwide remote diagnostics
> Logging of CAN data
> Visualization software with the possibility to configure on PC
> Data transmission to tablet and mobile phone
> Apps available

CANLINK® MOBILE

> CAN data transmission via EDGE/UMTS/HSDPA
> Robust and easy-to-install hardware
> GPS, GLONASS, Beidou
> Web-based fleet management and remote diagnostics
> Bidirectional Communication
> Firmware over the air (FOTA)

PROEMION GmbH
Headquarters | Donaustr. 14 | 36043 Fulda, Germany
Phone +49 661 9490-600 | info@proemion.com

MORE INFORMATION AT
WWW.PROEMION.COM

CAN FD Interfaces for High-Speed USB

PCAN-USB FD

Single Channel CAN FD Interface

- Adapter for High-speed USB 2.0 (compatible to USB 1.1 and USB 3.0)
- Time stamp resolution 1 µs
- CAN bus connection via D-Sub, 9-pin
- Complies with CAN specifications 2.0 A/B and FD 1.0
- CAN FD bit rates for the data field up to **12 Mbit/s**
- CAN bit rates from 40 kbit/s up to 1 Mbit/s
- Measurement of the bus load including error and overload frames on the physical bus
- Induced error generation for incoming and outgoing messages
- Switchable CAN termination and 5-Volt supply
- Galvanic isolation up to 500 V
- Extended operating temperature range from -40 to 85 °C

Scope of Supply for all CAN FD Interfaces

- CAN FD interface drivers for Windows 8.1, 7, Vista and **Linux** (32/64 bit)
- PCAN-View: Software for monitoring CAN and CAN FD busses for Windows (32/64 bit)
- PCAN-Basic: API for developing applications with CAN and CAN FD connection for Windows (32/64 bit)

PCAN-USB Pro FD

Dual Channel CAN FD & LIN Interface

- Adapter for High-speed USB 2.0 (compatible to USB 1.1 and USB 3.0)
- Time stamp resolution 1 µs
- Transmitting and receiving of CAN FD and LIN messages using two D-Sub connections
- Complies with CAN specifications 2.0 A/B and FD 1.0
- CAN FD bit rates for the data field up to **12 Mbit/s**
- CAN bit rates from 40 kbit/s up to 1 Mbit/s
- Measurement of the bus load including error and overload frames on the physical bus
- Induced error generation for incoming and outgoing messages
- Switchable CAN termination and 5-Volt supply
- Each CAN channel is separately opto-decoupled against USB and LIN up to 500 V
- Extended operating temperature range from -40 to 85 °C

LIN operation properties ...

- Bit rates from 1 kbit/s up to 20 kbit/s
- Both LIN channels (common ground) are optodecoupled against USB and CAN FD
- Can be used as a LIN master or slave (1 ms master task resolution)

www.peak-system.com

Take a look at our website for the international sales partners. Scan the QR code on the left to open that page.

PEAK-System Technik GmbH

Otto-Roehm-Str. 69, 64293 Darmstadt, Germany
Phone: +49 6151 8173-20 - Fax: +49 6151 8173-29
E-mail: info@peak-system.com

® PEAK System