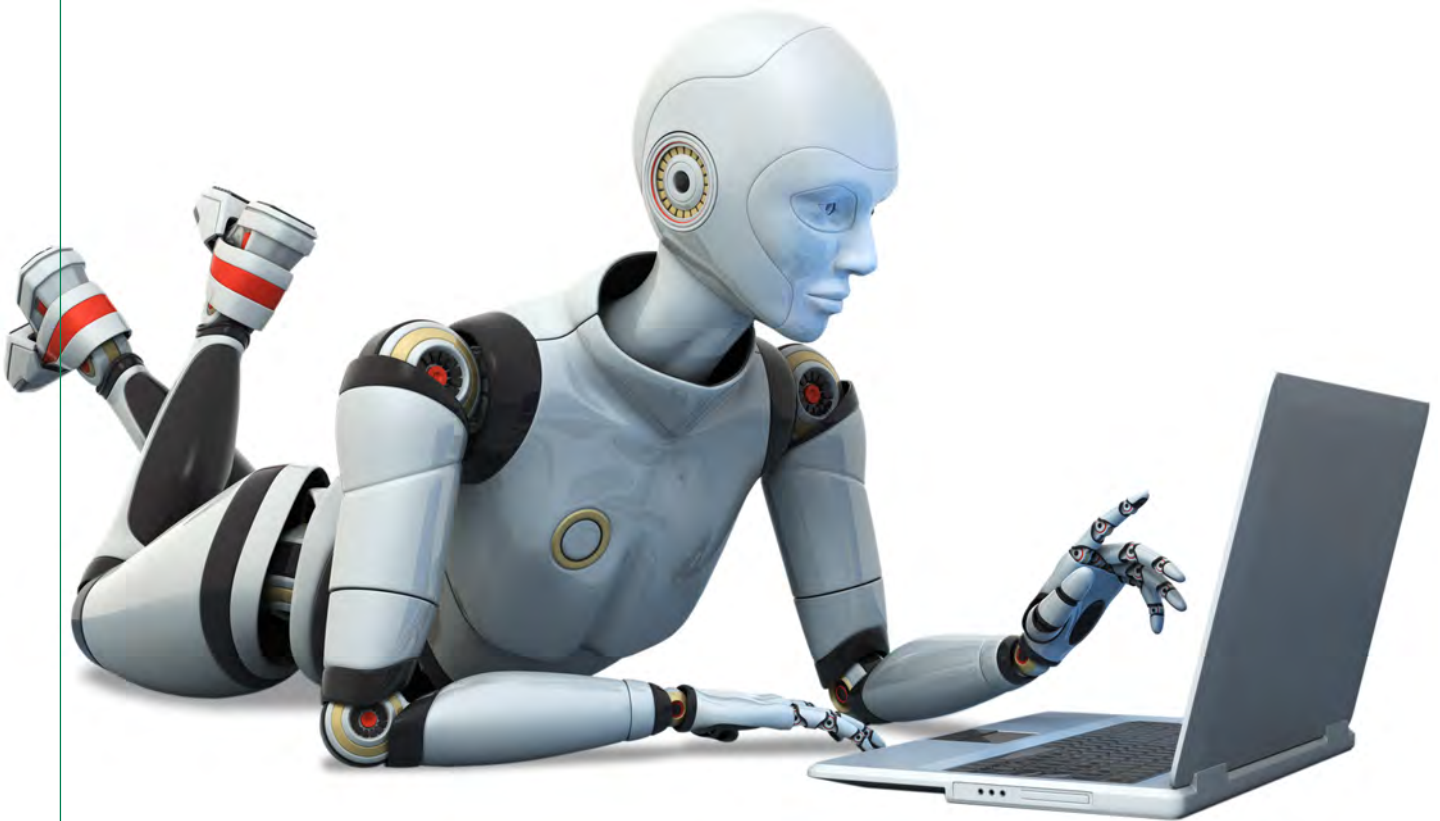


September 2015

CAN Newsletter

Hardware + Software + Tools + Engineering



CAN-based safety parameterization

Extending the ODX standard

From concept model to production code

ODX-based flash solution

Software engineering

www.can-newsletter.org



Hardware & Software for CAN / LIN Bus Applications

■ PCAN-Diag 2

Handheld Device for CAN Bus Diagnostics

The PCAN-Diag 2 is a mobile device to diagnose a CAN bus at the physical and protocol level.

- Clear CAN traffic representation in lists with configurable symbolic message representation
- Transmission of individual CAN frames or CAN frame lists
- Recording and playback of CAN traffic
- Optional automatic bit rate detection
- Bus load and termination measurement
- Internal memory with USB connection for saving projects, screenshots, traces, and CSV files

Oscilloscope functions

- Oscilloscope with two independent measurement channels, each with a maximum sample rate of 20 MHz
- Display of the CAN-High, CAN-Low, and difference signal
- Configurable trigger events: frame start, frame end, CAN errors, or individual CAN frames based on their CAN ID

■ PCAN-MiniDisplay

Visualization and Recording of CAN Data

The PCAN-MiniDisplay is used as a human-machine interface for the visualization of CAN data. In addition to the built-in PCAN-MiniDisplay there is a version in plastic casing with push buttons available.

- CAN connection via a High-speed CAN and a Single-wire CAN channel (ISO 11898-2, SAE J2411)
- Wake-up function via CAN
- Slot for microSD memory card (max. 32 GByte)
- USB port to access the memory card using a PC
- TFT display with 320 x 240 pixel resolution
- Freely configurable visualization of CAN data via text files
- Running configurations from the memory card
- Recording of incoming CAN messages to the internal memory card (optional filtering of CAN IDs per channel)
- Operating temperature range from -20 to 70 °C
- Dimensions: 70 x 50 mm (built-in version)
- Voltage supply from 7 to 30 V



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



Software engineering

CAN-based safety parameterization	12
Extending the ODX standard	16
From concept model to production code	26
ODX-based flash solution	42

Imprint

Publisher

CAN in Automation GmbH
Kontumazgarten 3
DE-90429 Nuremberg

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

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

CEO Holger Zeltwanger
AG Nürnberg 24338

PDF subscribers: 3300

Editors

pr@can-cia.org

Annegret Emerich
Cindy Weissmueller
Holger Zeltwanger
(responsible according to the press law)

Layout

Nickel Plankermann

Media consultants

Julia Dallhammer
Gisela Scheib
(responsible according to the press law)
Meng Xie

Distribution manager

Julia Dallhammer

© Copyright

CAN in Automation GmbH



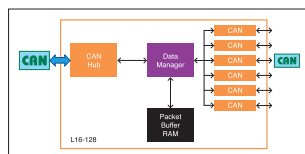
Applications

Controller with universal bus connections	24
Squirreling away solar energy for winter	30



System design

CAN frames through IP networks	4
CANopen in series production	8
Hybrid CAN and CAN FD networks	20
CAN FD: from theory to practice	34



Device design

Flexible and scalable CAN solutions	38
-------------------------------------	----

New CiA website is online

In August, we went online with a newly designed [website](#). The new site has a more modern look and feel than the old one and hopefully it is also more clearly arranged. While there are still some bugs we have to iron out, we welcome any [feedback](#) you might have. Anyway, we also hope you enjoy this issue of our CAN Newsletter Magazine and if you want to read even more articles, you can check out our [CAN Newsletter Online](#).

CAN frames through IP networks

Taking CAN data frames and envelope them into Ethernet IP packages is an interesting opportunity. If you can configure the needed gateway, it is even better.

Imagine several production lines, all operated with CAN networks to transfer measuring and control data. They may be located in different halls. For a direct supervision, the data must be gathered at a central point. An office across the street is intended as a control center to obtain a comfortable remote diagnosis. These are long ways – too far for CAN networks.

CAN is local

The CAN network is designed for a localized system. Initially used in cars, it nowadays can be found in whole production facilities. This implies a greater amount of data that can only be handled with appropriately high data rates. On the other hand, this results in a shorter maximum CAN network length. As a rule of thumb, a CAN network may have a maximum length of 40 m if a bit-rate of 1 Mbit/s is used. This isn't much if several I/O nodes within a machine must be reached. Creating a network of far-reaching facilities is especially challenging. The aim is to make measuring and control data available live at remote locations. Possible tasks are the exchange of data between remote CAN networks or, as already mentioned, the comfortable surveillance of the whole CAN network from a control center.

CAN-over-IP helps

The starting point here is the almost ubiquitous IP world. Why not take advantage of a cheap and possibly already existing IP network for CAN data? This is the point where the PCAN-Gateway family from Peak-System comes into play. A gateway connects the CAN network with the IP network. It takes the CAN frames, wraps them into IP packets, and sends them through the LAN. Another gateway, which is attached to a remote CAN network, unwraps the received packets and transmits the contained CAN messages on the CAN network. The CAN frames are tunneled unchanged and then replayed one by one. The PCAN-Ethernet Gateway DR, the initial member of the PCAN-Gateway family, does this via a customary Ethernet LAN.

A side aspect of linking several CAN networks via CAN-over-IP is the possibility of using different bit-rates. Apart from the correct bit-rate being used by a gateway on the connected CAN network, no further adaptations are needed for the CAN frames. They are simply transmitted on the remote CAN network in its speed. As a side note, this speed conversion is also possible between two CAN networks that are connected to the same PCAN-Ethernet Gateway DR.



Gateway configuration

A gateway is configured via a web interface, just like a common Internet router. Besides the basic settings for IP networks, the configuration for the high-speed CAN channels, for the so-called routes, and for filter settings can be found here. The information exchange between gateways is done through message forwarding via routes. For data transfer in one direction, an outbound route must be defined in gateway A with gateway B as the destination. Gateway B must be set up to accept messages from this (inbound) route. When a route is initialized, the gateways do a handshake and establish an additional channel for supervising the communication.

Because a gateway can be configured with up to eight outbound or inbound routes, more than two gateways can communicate with each other in an IP network. Thus, there is no restriction to a 1-by-1 connection. In case of the remote maintenance example, the gateway in the control center receives the CAN frames via LAN from the gateways attached to the machines and puts them onto the local CAN network. In turn, different analysis devices can be connected to this CAN network, e.g. displays, switches, or a PC with a CAN interface.

Not all CAN messages are necessarily relevant for other CAN networks. Therefore filtering can be set up in a gateway to accept the CAN messages by CAN ID ranges ▶

CAN FOR EXTREME ENVIRONMENTS

CANopen Coupler D-Sub /XTR, 750-338/040-000

CANopen®

XTR



750-338/040-000



750-8204



750-337, -837



750-337/040-000
750-837/040-000



750-338



750-347



750-348



767-1501



767-658 (CAN)

The WAGO-I/O-SYSTEM 750 XTR – TAKING IT TO THE EXTREME

- eXTReme temperature ... from -40°C to +70°C
- eXTReme vibration ... up to 5g acceleration
- eXTReme isolation ... up to 5 kV impulse voltage
- eXTReme dimensions ... as compact as 750 Series standard

www.wago.com

**WE
INNOVATE!**

WAGO®

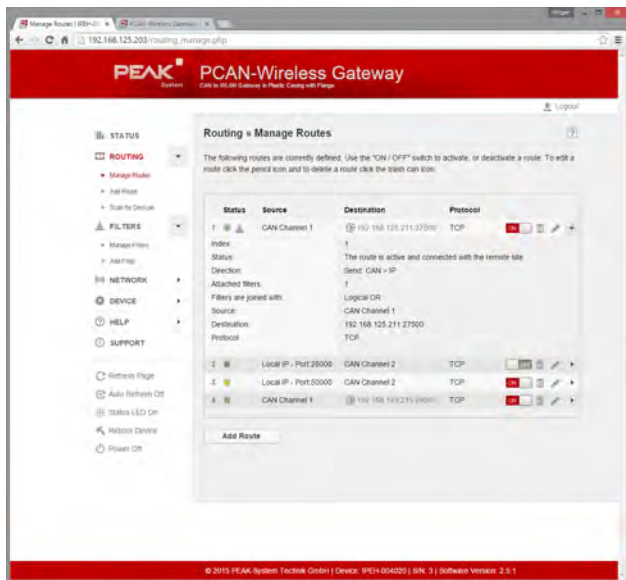


Figure 1: In the web interface of this gateway, two outbound routes and two inbound routes were defined; CAN messages coming via the IP network are accepted on port 50000 (Photo: Peak)

or to CAN ID masks. Different filter definitions can be applied to specific outbound routes of a gateway. The CAN network that receives CAN messages over IP benefits of not being flooded with the whole CAN traffic of one or several remote CAN networks and also the performance of CAN-over-IP is gained.

Reliability aspects

When configuring the routes, the IP protocol can be chosen: either TCP or UDP. While TCP ensures the transmission and the reception of data packets through the feedback of the specific recipient, UDP lacks such a mechanism. Instead, the latter has the advantage of a lower overhead. In this case, a safety-related communication requiring a reliable transmission is not recommended. Here, the closed CAN network continues to play out its trump card.

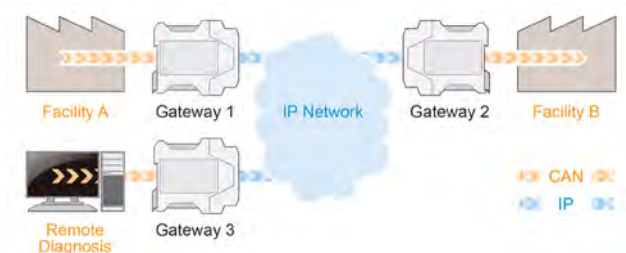


Figure 2: Example of a network of two facilities with a supervising instance (Photo: Peak)



Figure 3: A Virtual PCAN-Gateway in the PC permits the direct access to the CAN traffic in facility A (Photo: Peak)

Diversity of transmission ways

Besides the PCAN-Ethernet Gateway DR, there are also gateways with wireless IP communication available. Those can cover the transmission ways that cannot be bridged by cabling. Moreover, tunneling may not necessarily take place between two hardware gateways. The Virtual PCAN-Gateway software can be installed on a Windows PC. It can directly communicate with the PCAN-Gateway hardware via an IP network that is connected to the PC. The linking between the PC and a gateway is done with routes, like the linking between two hardware gateways. However, the remote CAN network is handled like any other CAN network being directly connected to the PC. Therefore, all programs on the PC that can communicate with the PCAN environment (based on drivers and APIs from Peak-System) are able to access this remote CAN network in the usual way.

The scenario of virtual gateways is not restricted to PCs. Mobile devices, e.g. tablet computers, are also able to act as counterparts to PCAN-Gateways via custom applications. This results in new options for a quick access to CAN networks, for example for field technicians.

Regarding the transmission reliability of the CAN network, the CAN frame transport through an IP network cannot reach the same level. But this is not the relevant point for remote maintenance. Instead, distances that cannot be bridged by CAN networks and the easy availability of data are paramount. These requirements can easily be realized with PCAN-Gateways.



Author

Mark Gerber
Peak-System Technik GmbH
www.peak-system.com
info@peak-system.com



Look what we have for you!

Smart 3D sensors for mobile machines

Fast detection of 3D scenes and automatic object recognition.

Simple application solutions thanks to 3D data preprocessed in the sensor.

Patented PMD time-of-flight technology for quick distance detection with ranges up to 35 m.

Optimised for reliable outdoor use with IP 67 and IP 69K.



www.ifm.com/gb/o3m

Phone: +49 0800 16 16 16 4

CANopen in series production

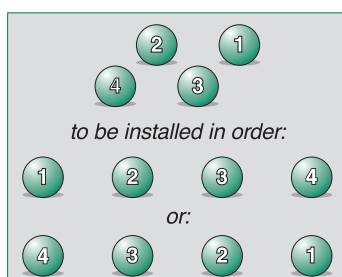
The idea of digitalization is to adopt new processes enabled by more sophisticated technologies. It should not include trying to force new technologies to operate according to ancient, often work intensive, processes.

Traditional industrial control system installations where repeatability is not an issue are rare. In the production of mobile machines and special vehicles, configurations must be repeated dozens to thousands of times for each type of device, targeted into various systems and positions. Another main difference is that industrial control systems may be running for months or years after each start-up, but mobile control systems are started and stopped when the machine operator turns the ignition key on and off. Therefore, system start-up time is critical and in order to keep it minimum, device configurations must be stored into the devices in order to avoid massive downloads during system start-up.

From a producibility and serviceability point of view, there must always be second sources for the primary device, in order to guarantee the continuous availability of devices. In practice, if system vendors do not supply the spare parts, service personnel usually finds a way to arrange them by themselves. On the one hand, CANopen enables system vendors to limit supported devices, but in another hand, CANopen also enables the easy management of second sources.

The main challenge in the series production of many systems is that there are numerous instances of the same type of device. Even I/O-extension devices provide a simpler structure when there are more units containing less I/Os, for interfacing sensors and actuators located close to the unit. A single thin network cable is simpler than a mass of long I/O cables. In order to keep logistics simple and inexpensive, typically only a few items are kept in storage, which are adapted to as many positions as possible by changing the configuration.

Before starting the actual configuration download, the grand challenge is to identify each target position unambiguously and achieve correct mapping between target positions and corresponding configurations. The fact is that a uniform mechanism, by which the system itself could unambiguously detect the target positions in a general case without constraints, does not exist. The traditional approach of assigning the node-ID and bit-rate in connector pin connections or coding plugs has been used, but that principle is based on the least dependable domain of mobile control systems: cabling [1].



When cabling is used for the assignment

of the most essential parameters, even small failures can prevent the use of the system. Constant and good quality and repeatability are the main targets.

Therefore, any fancy mechanism, such as using network propagation delay as an aid for detecting the device locations, cannot be accepted. One of the main features of CAN – monitoring the resulting states of transmitted bits – actually guarantees reception synchronization within a single bit-time and prevents the reliable use of propagation delay based node-ID assignment. Furthermore, any deviation in the cabling changes the signal propagation time and slew rate, which can corrupt the propagation delay based assignment. Different dimensions and layouts among product variants also limit the usability of propagation time based node-ID assignment.

The main concerns in CANopen systems, before other settings, are node-ID and bit-rate. It has already been described in the literature [1], how the download process can be integrated into the logical extension of the design process. This article starts by reviewing the bit-rate and node-ID assignment. Next, various aspects of plug'n'play are reviewed. Then a generic plug2play approach, without any special constraints, is described in detail. A discussion of some special topics follows the presented approach. Finally, the conclusions are described.

Importance of the bit-rate

The bit-rate is the most important parameter, because invalid settings prevent the entire network communication. The use of mechanical elements for setting the bit-rate has the same risks as setting the node-ID has, as described in the next section [1]. In general, the use of mechanical elements for configurations means the use of human labor for the work, and human mistakes. The efficiency and average quality of human beings is not as good as that of automated work. Furthermore, possible changes in the switch layout or type require new instructions distributed to the field. Changes will probably be introduced, due to the obsolescence of devices.

An automatic detection of the bit-rate sounds nice as long as it works without problems. But, at least one device in each network must have a fixed bit-rate, which can be used as a reference by the other devices. If there are only devices waiting for any valid bit-rate, communication never starts. A further risk of an automatic bit-rate detection is that if the device defining the bit-rate is changed and the new one has an invalid bit-rate, everything seems to work but, e.g. due to the overall network length or

required bandwidth, communication at the new, unintentionally set bit-rate is unstable and makes the system useless.

Node-ID

The node-ID is the second important parameter in each CANopen device. Many engineers would like to have plug'n'play systems without understanding the basic concept. First, the node-ID defines priorities of the basic management protocols – boot, heartbeat, and SDO – of each device and thus a random assignment of the node-IDs can break the intentionally designed communication schedule and in the end make the system unstable.

Another consequence is that using random node-IDs makes basic system monitoring functions useless by changing fixed physical to logical mapping, which is defined during the design process [2]. Structural monitoring is one of the most important services in each system, especially in safety relevant control systems, where the control applications need to monitor the system structure continuously [6].

[Assignment methods of node-IDs](#) have already been reviewed [1] and readers are advised to read the earlier article in order to get more details. One issue is missing from that article: In the described approach, where existing connections or switch position patterns are used for determining the node-ID, there is no guarantee that the existing configurations are correct. The approach also prevents the efficient use of 2nd source devices and upgraded to new models with different mechanical structures.

Plug'n'why-is-this-not-working

Many engineers are interested in full plug'n'play – a concept where everything can just be put together and the system works without any parameterization. Such a concept can be achieved, but only by defining tight constraints. In the traditional industrial approach, the bit-rate and node-ID have to be set – most often manually, by electricians – before the start-up of the system. Then, as part of the start-up, the control system sets the other parameters by relying on the fact that correct devices were installed with correct node-IDs into correct locations. Therefore, plug'n'play systems quite often turn out to be plug'n'why-is-this-not-working systems.

If each type of unit was only used once, each unit could have a fixed node-ID and everything could just be connected. But first, the supported devices would be very limited or there would be a serious risk of inconsistency. Second, it would not be possible to utilize the entire CANopen ecosystem, because using generic devices typically leads to an efficient functional re-use by using multiple instances of the same type of devices. Third, the structure of such a system is very limited and extensions sooner or later lead to a maintenance and service nightmare, due to the ever increasing number of different types of spare parts.

One intermediate approach is to utilize the standard devices with fixed configurations. This is also a maintenance and service nightmare, because there would be dozens of equal devices in the warehouses having different configurations, which cannot be visually identified. ▶



THE PFC200 CONTROLLER

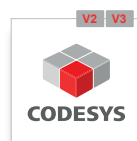
Compelling, Fast and Intelligent

CANopen®



- High processing speed
- Programmable with CODESYS 2 and e!COCKPIT (based on CODESYS 3)
- Configuration and visualization via Web-server
- Integrated security functions
- Robust and maintenance-free

www.wago.com/pfc200



WE INNOVATE!

WAGO®

The only advantage may be achieved via larger purchase lots. The devices are still generic and configurable, which will lead to misinterpreted device IDs due to the same outline. Another risk is that the same device could be obtained outside the authorized supply chain, without properly predefined configurations.

The world is continuously evolving and devices are too. Thus, older devices become obsolete and new devices are introduced to the market. If constant or application managed configurations are used everywhere, it is a heavy task to handle the evolving devices in all existing product variants. It does not matter if one uses off-the-shelf or full-custom devices: a requirement of one-to-one equality always leads to the slow development or very high upgrade costs.

Functional safety requirements introduce interesting new aspects to the problem. In principle, risk reduction and managed processes must cover each phase of the systems' life cycle. Thus, it is important that any kind of risk of spare part changes is carefully analyzed and found risks are eliminated. It is also necessary to keep the systems consistent over their entire life cycle, which requires good assembly and service processes, in addition to the design process.

CANopen supports plug2play

Instead of the traditional plug'n'play approach, CANopen supports the plug2play approach better. The main idea of the plug2play approach is that each device is always processed equally, independent of whether it is a new one or borrowed from a neighboring system. The first plug occurs when a device is plugged into the configuration tool, where its identity can be checked and the appropriate configuration can be downloaded, stored, and verified. The second plug occurs, when the approved and configured device is plugged into its target position, without fear that something fails. If the two plug operations are successful, the play is a fact. Based on the practical experience that more failures occur during each disassembly and assembly action, the most important effect of plug'n'play is that problems appear before the device is installed into its target location. Thus, the correction is cheaper and faster than it was before.

The download process starts with a detailed identification of the device and only configurations dedicated to the found device are provided. CANopen provides a perfect mechanism for an unambiguous device identification based on device type, vendor-ID, product code, and revision number. The serial number can be stored in order to enable a detailed structural follow-up. Based on the detailed identification, available configurations stored as DCF files are provided to enable the user to select the desired target location. After selecting the location, the download of pre-defined parameter values is followed by store, reset, and verify. The verification improves not only the functional safety integrity level, but also ensures the success of the download phase.

The main advantage of the plug2play approach is that it is based on tools external to the control applications. In other words, there is no need for application re-testing and

re-certifications due to a change of a single device. Generic tools can be used with any CANopen device and with systems with any set of CANopen devices. Furthermore, if other integration frameworks are also used, all tools can be maintained as a solid toolkit.

Another advantage is that each device is verified during the first plug, just before installation into the target system. This characteristic minimizes the wasted time in case of an invalid device, because it has not been installed into the target location yet. While the download is performed only if the device and the corresponding data file contain the equal identity, it is impossible to get invalid configuration downloaded into an invalid device. System misbehavior caused by an invalid device and/or configuration cannot be the result, because the validity of both the device and configuration are approved during the process [1]. The only error source could be parameter values which are not optimal for this particular position, which is not within the scope of this article. There are well-known and proven approaches for helping the parameter value assignment [3].

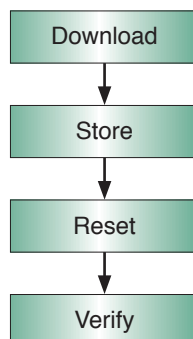
The use of DCF files (or XDC in the future) for the information transfer from design to assembly and service guarantees that information is not lost in any phase. In EDS files, device vendors are enclosing meta information for parameters, which can be used for controlling the download process. The access type defines the absolute access right of each object and the lowest bit of object flags can be used in order to prevent a tool from a download [1]. Such an approach keeps the process independent from individual tools, tool vendors, and versions.

Device parameters

One special topic of parameterization is managing which objects are accessed during the download phase through the object access type and object flags [1]. Such a mechanism enables e.g. overwrite protection of the factory calibration parameters set by the device vendor. The mechanism can also be utilized for the protection of other parameters, if required. Object flag values of EDS files may be adjusted by system integrators e.g. during the device acceptance test, before accepting it into a device library.

When using CANopen devices, an alternative path can be taken just by adding an alternative device, which conforms to the same device profile as the original part and supports the signal and parameter objects in use [3]. Usually, the values of some of the device parameters need to be adjusted in order to get the new item into the network [4]. In the optimal case, the device parameters that are specific to the device profile can just be imported from a DCF file of the original device into a DCF file of the new device [3].

CANopen device profiles enable functional scaling among device profiles by supporting common default data types [5]. Such characteristics enable the scaling of the functional complexity by changing the devices with corresponding configurations. The key issue is that in the corresponding control applications no changes are required. In such a functional scaling, the number of devices can also vary among the options [7], which can be solved by adjusting the slave assignment of the NMT-master and the application object values affecting the usage of signals from the heartbeat consumer. ▶



Some parameters in some devices can be adjusted by e.g. in modulo-2, -4, -8, etc. steps. With such devices, the download-save-reset-verify process reveals if parameter values do not follow the correct modulo. Typically, attempts to download values below the minimum and above the maximum allowed value results in an error indication. A more complex example of tracing inconsistencies between two or more parameters occurs when e.g. a 16-bit pressure signal is mapped into a pressure transmitter's TPDO and too many decimal digits are set for the signal in either the physical unit or decimal digits parameters. They may exist in a DCF file, but cannot be downloaded to a device.

In CANopen, firmware or application software is considered a value of one parameter [1] [8]. If the device supports the standardized SW download mechanism, it can be updated during the same download transaction with the other parameter values. The information source is based on the DCF file, where a link to the actual compiled application exists. Thus, there are no limitations in the file format from a design process' and tool's point of view. However, the format supported by the target device must be used for storage. Configuration download tools simply transfer the file contents as a domain type value. Configuration management is a tool that enables the efficient re-use of fixed function devices. CANopen contains native support for comprehensive configuration management. In conjunction with the device profiles, configuration management can be utilized for device replacements and functional scaling, without a mandatory need for changes into control applications.

CANopen provides a good support for the series production of the systems. Instead of plug'n'play, which only works sometimes under special conditions, plug2play – working always and without constraints – was found more productive. It enables the flexible use of alternative devices in order to maximize the spare part availability from several part vendors, but also enables system integrators to keep total control over official spare parts.

The use of local storages enables fast system start-ups, because parameters are not downloaded on start-up. Off-board management processes decrease the amount of human work and thus human mistakes. They also provide divert part identity verification and early notification of failures. Independence from the control applications minimizes the need of application changes and re-certifications caused by ever updating library devices. In addition to the system devices, commitment to the standardized mechanisms provides tool independence. An option for 2nd source tools provides a back-up for continuous assembly and service. Commitment to the standardized, well-documented, and proven process supports the design work with safety relevant systems. ◀

Author



Heikki Saha
TK Engineering
www.tke.fi
[References](#)

CANopen extension for SIMATIC® S7-1200



IXXAT

CM CANopen

Communication module for connecting CAN-based field devices with the SIMATIC® S7-1200 world

- Comprehensive CANopen functionality for master or slave mode
- Transparent CAN 2.0A mode for the support of alternative protocols
- Easy PLC programming within the TIA Portal using pre-programmed function blocks
- Intuitive Microsoft Windows application for the CANopen network configuration included

With CM CANopen HMS offers under the brand IXXAT a module for the easy integration of CANopen and CAN-based I/O modules, drives or sensors into SIMATIC S7-1200 controllers as well as in PROFIBUS and PROFINET networks.



Also available: **1 SI CANopen**
CANopen module with CAN 2.0A support for
SIMATIC ET200S decentralized peripheral systems

SIMATIC, STEP 7, TIA Portal and images of the S7-1200 and ET200S are intellectual property of Siemens AG Germany and copyright protected.

HMS Industrial Networks GmbH
Emmy-Noether-Str. 17 · 76131 Karlsruhe

+49 721 989777-000 · info@hms-networks.de

www.anybus.com · www.ixxat.com · www.netbiter.com



CAN-based safety parameterization

The development and maintenance cost of safety related software packages is much higher than for non-safety functions. Using parameterization, software components can be standardized across multiple machine variants.

Using a suitable method, the machine designer can adapt software components to the particular characteristics of a machine configuration without modifying safety relevant software to minimize the certification and maintenance effort for safety relevant modules. Parameterized programming is often mentioned as a technique for a reliable reuse of software. In this technique, modules are parameterized over very general interfaces that describe required properties of an environment for the module to work correctly. The reusability of the software may cut down costs of the rising demands on the flexibility and reliability of software. “The basic idea of parameterized programming is to maximize program reuse by storing programs in the most general form possible. One can construct a new program module from an old one just by instantiating the relevant parameters.”

Use-cases of a parameterization solution for mobile systems

Figure 1 shows functional use-cases of a mobile system from different perspectives. All use-cases need operations that involve “safety parameterization” in the system. The application that uses the parameters runs in the electronic control unit (ECU), which can be seen as the brain of the mobile working machine. A solution must allow the ECU application developer to define parameter sets during the development phase of the ECU application. It must also enable the operator to read and write individual parameter values from a personal computer (PC) tool or by using a non PC-based human-machine-interface (HMI). The ECU application should use parameter values stored in the ECU’s memory, too.

Available standardized parameterization tools do not fulfill the requirement of writing individual parameter values in the field, which is a special use-case for mobile applications. For instance, the mobile machine needs parameterization during the initial operation for adjusting sensors and actuators. In addition, it might be necessary to exchange components during a preventive maintenance. The PC tool-chain Kefex is used as an example to demonstrate different tasks during the development and maintenance life cycle of an ECU application. Its first integral part is the tool RAM-View, which supports the use-cases for parameterization.

Software-based safety parameterization

The component RAM-View is used to read and write parameters and diagnostic variables from or to the ECU’s memory and supports configuration and monitoring. It can be used to create and edit parameter sets and to view or mod-



Figure 1: Use-case for parameterization in a mobile machine

ify the variables on the ECU, too (see Figure 2). The Kefex client, which also runs on a PC, provides components for communication with the ECU and displays customer defined values on the HMI. Different dynamic link libraries (DLL) support CAN interfaces such as from Peak, Vector, Ixxat, and CPC-PP to establish the communication between the PC and the ECU.

To work with variables defined in RAM-View, it is necessary to run the Kefex server on the ECU side. The server provides mechanisms to read and write memory contents from and to a PC tool and reads parameter values from the ECU hardware abstraction layer (ECU HAL). The server supports working with multiple projects on one ECU, too. These multiple RAM-View projects can be linked with one application. Now, how can RAM-View become a tool for safety parameterization of safety relevant software modules developed in either the safety variant of the programming language “C” or IEC-61131 (Codesys Safety SIL-2)?

Safety parameterization techniques

Both safety relevant standards EN ISO 13849 and EN 62061 describe the same requirements towards software-based safety parameterization. RAM-View fulfills these requirements. As a supplier of 32-bit safety ECUs like the [ESX-3XL](#) or [ESX-3XM](#), Sensor-Technik Wiedemann (STW) provides the dedicated software tool RAM-View Safety for safety parameterization. It delivers appropriate actions to verify the tool configuration and prevent unauthorized modification with password features. The required measures to control valid values are assured by the cooperation between the Kefex client and server.

A suitable action to handle the data corruption of single parameters before transmission is reading the values and confirming their validity through the operator. To control the effects of errors arising from the parameter transmission ►

process, incomplete parameter transmission, and the effects of faults and failures of the hard- and software, the Kefex server verifies checksums. Kefex RAM-View Safety fulfills all requirements of software-based safety parameterization realized by a suitable special procedure. This includes the confirmation of input parameters to the safety ECU by re-transmission of the parameters to the parameterization tool. The procedure also includes a confirmation by an operator and an automatic check. Diverse functions avoid systematic failures. These functions cover encoding/decoding within the transmission/retransmission process and visualization of the non-safety and safety-related values to the operator.

Interactions between RAM-View and ECU

The system is designed in such a way that safety critical decisions are either made by a safety ECU software or are covered by a clearly defined parameterization process. PCs cannot make any safety related decisions. Therefore, although the PC calculates all checksums, the decision about the correctness of these checksums is only made by the safety ECU. The variable description tables exported by RAM-View are linked with the ECU application. RAM-View therefore is a "T3" off-line support tool as defined in DIN EN 61508-4. This standard was applied to the tool qualification process. As a further defensive measure, the Kefex server performs consistency checks on the exported data.

The whole communication between the PC and the ECU is considered a "black channel". It includes writing the payload data to the ECU's memory. The server ensures that accidental accesses from RAM-View or other PC tools do not affect safety critical parameters. This is achieved by using the memory protection mechanisms implemented within the ESX-3XL and ESX-3XM. By utilizing the mechanism, the recognition of an accidental manipulation of safety data can be ensured and the PC tool does not need to be considered as an online support tool according to DIN EN 61508.

Process for creating and writing pre-defined safety parameter sets

The following process is used for creating and writing safe parameter set files. It can be used to pre-define parameter values for deployment to a number of ECUs, e.g. end-of-line programming in series production. The following steps are performed for creating parameter set files on the ECU:

1. configuring desired parameter values,
2. reading and storing values of selected parameter lists including their checksums,
3. checking the created parameter set file.

When writing parameter set files to the ECU, the following steps are performed:

The PC tool

1. checks the checksums of the parameter values and fails if they do not match,
2. writes the contents of the parameter values to the ECU's memory,
3. and sends checksums of the new data.

The ECU then checks whether the received checksum matches the received data and addresses. An operator reading back the values and confirming their correctness ▷

USB-to-CAN V2 The good just got better!



The new generation of IXXAT CAN Interfaces

- For mobile analysis and configuration of CAN systems as well as sophisticated simulation and control
- Up to two CAN interfaces (optional low-speed CAN and LIN)
- USB 2.0 Hi-Speed for minimal latency and high data throughput
- Drivers for Windows and Linux included

The latest generation of IXXAT USB/CAN interfaces from HMS is even more powerful and versatile – at a really low price. The interfaces are available in different versions (compact, professional, automotive).

HMS supports all versions with analysis and configuration tools as well as with drivers, e.g. for CAN, CANopen and SAE J1939.

NEW

Also available as build-in version (with or without slot bracket) for direct integration into computers or customer devices



HMS Industrial Networks GmbH
Emmy-Noether-Str. 17 · 76131 Karlsruhe

+49 721 989777-000 · info@hms-networks.de
www.anybus.com · www.ixxat.com · www.netbiter.com

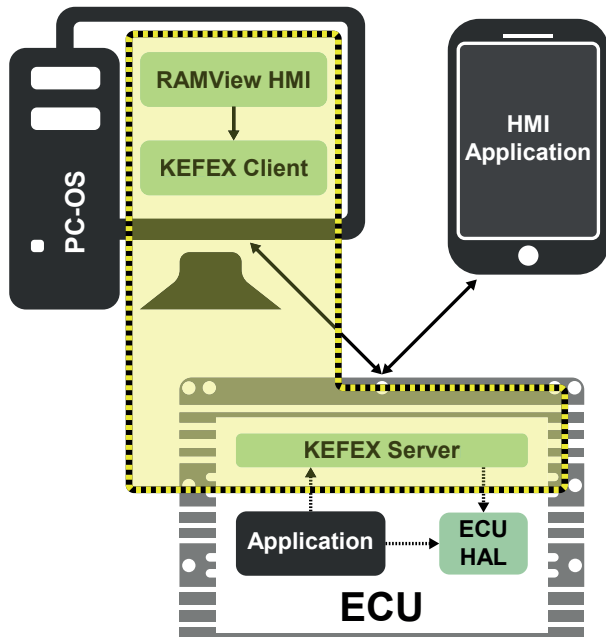


Figure 2: Relevant components for safety parameterization

is not required as the consistency of the data stored in the file is ensured by checksums.

Process for parameter modification

The following process is used to change one or more parameters when there is no pre-defined parameter value file. This could happen during the modification by a service technician in the field who needs to set individual parameter values from a PC or HMI.

The PC tool or HMI

1. writes the parameter values to the ECU,
2. reads back the parameter values from the ECU,
3. reports read back values to the application layer HMI,
4. waits for approval by the operator,
5. calculates the checksums,
6. and sends checksums of the new data values.

The ECU then checks whether the checksums received match the received data and addresses. This procedure prevents undesired changes in the data during the transmission between the PC and ECU memory. The transmission and presentation paths are diverse and allow detection of systematic failures. The parameter values are checked to be in a valid range and hardware faults in the memory are detected. Of course, if any part of the data was changed on the way, an incorrect checksum will identify this.

Independence by multiple instantiation

A clear separation between parameters that are used for safety critical functionality and parameters that are used for non-interfering functionality is achieved by the Kefex server. It allows multiple instantiations with separate sets of parameter definitions.

Two (or more) different RAM-View projects, each containing a parameter set definition, can be defined with parameters used for safety or non-safety functionality. This

approach leads to a clear separation between safety-relevant and non-safety-relevant parameters in:

- ◆ The project part on a personal computer: the project with the safety-relevant parameters does not need to be touched when one is only changing non-safety relevant parameters,
- ◆ The ECU volatile memory at run-time, which utilizes the memory protection mechanism,
- ◆ The ECU code, which entails separate variable description tables; it is not necessary to change the file with safety-relevant parameters if only the non-safety relevant parameters are changed.

As a result, the user will see a reduced testing effort as long as only non-safety relevant parameter sets are changed.

Conclusion

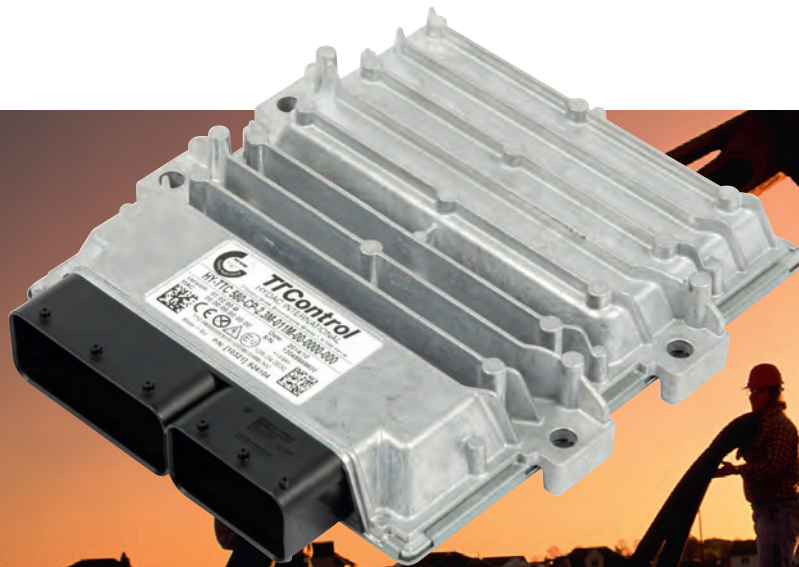
Kefex RAM-View Safety fulfills the standards of EN ISO 13849-1 and DIN EN 62061 for safety based parameterizations. The solution is designed in such a way that all safety critical decisions are either made by the safety-related ECU system or are covered by its well-defined interaction process between the PC tool and the safety ECU. As a result, the use-cases shown in Figure 1 can be covered. RAM-View provides functionality for the definition of parameter sets during the development phase of the ECU application. The processes for creating and writing the parameter set files prevent undesired changes in data by storing the data as a black box on the PC side. If any part of the data is changed on the way, the ECU detects this by controlling the checksums.

The operator can use a process for parameter modification, which ensures that no undesired changes occur in the data during the transmission between PC and ECU memory. The transmission and presentation paths are diverse and allow the detection of systematic problems. The ECU application uses parameter values stored in the ECU's memory. A multiple instantiation supported by RAM-View achieves a clear separation between parameters that are used for safety critical and non-interfering functionality. The result is a reduced testing effort as long as only non-safety relevant parameter sets are changed. Kefex RAM-View is provided to be used in "C" applications and is designed to be integrated into a Codesys Safety SIL-2 run time system. ◀



Author

Kai Niestroj
Sensor-Technik Wiedemann
www.sensor-technik.de
info@sensor-technik.de



Powerful Control Units for High-Safety 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 easy 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
- Safety companion and safety mechanism in hardware

Connectivity

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

Performance

- 32 bit / 180 MHz TI TMS570 dual core lockstep processor (ARM architecture)
- Up to 2.3 MB RAM / 11 MB Flash
- Floating-point-unit

Robustness

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

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



Safety Certified ECUs



General Purpose ECUs



I/O Modules



Safe I/O Modules



Operator Interfaces

Extending the ODX standard

Limitations in the ODX standard make it difficult for files to be portable among different manufacturers. We make a case for the extension of the standard and its use in the development of ECUs through ODX-based code generation.

ODX has proven to be a useful tool for describing the communication of ECU diagnostic information. It has been adopted by various OEMs in different industries. In the past, manufacturers in the passenger and commercial vehicle market used non-standard methods to describe diagnostic ECU communication, including proprietary and non-machine-readable formats such as text documents or spread sheets [1]. As a result, information had to be manually implemented in the tools used in the machine's life cycle – a method which was prone to errors and was time and cost consuming [1].

Developed by Asam, the MCD-2 D standard (known as Open Diagnostic Data Exchange or ODX) was created to address these challenges. ODX is an XML-based, machine-readable data format created to specify and exchange vehicle and ECU diagnostic information including variants throughout the vehicle's life cycle [1]. It was created to be used in a standard-based software architecture for vehicle diagnostic communication including the Asam MCD-3 D and ISO 22900-2 (D-PDU API) Interfaces. ODX offers a method for defining communication that permits seamless processing of diagnostic, configuration, and flash reprogramming data. Standardizing diagnostic communication allows for reusability and thus helps to reduce errors, development time, and cost [1].

The current version of the ODX standard (V 2.2) supports several vehicle diagnostic protocols including KW 2000 (ISO 14230), and UDS (ISO 14229) [1]. It has proven to be a useful tool and it has expanded into other markets such as off-highway vehicles (agriculture, construction, forestry, etc.) as well as other non-vehicle markets.

Current uses and limitations

As more manufacturers adopt the ODX standard, its boundaries are pushed by their different applications and use-cases. Accordingly, the ODX standard has gone through several versions, which have expanded its capabilities since its public release in 2000. The standard has been extended to include a model for ECU flashing, vehicle network and identification information, additional communication parameter information, and support for variant coding and functional diagnostics [1]. However, several limitations still remain.

An example of the limitations of ODX is its support of only a few CAN protocols. Current diagnostic systems in the market are able to handle a variety of protocols, some of which are not specified in the standard. Figure 1 illustrates a generic ODX-based diagnostic system with

multiple protocol handling capabilities. In this system, the diagnostic application tells the D-server layer (MCD-3D) which ODX and comparam file to use to communicate with a specific ECU/data source. Communication parameters or 'comparams' are defined in the ODX standard and serve to specify the timing and logical behavior of diagnostic communication; these parameters are protocol-specific [2]. The standard provides comparam definitions for a few protocols but is missing the definition for other commonly used protocols such as CANopen, or non-diagnostic protocols used to access data from non-ECU sources such as databases, webservices, etc.

Consequently, designers of diagnostic systems manually create comparams to use ODX with other protocols. Since these parameters are not defined in the ODX standard, ODX files created for other protocols might not be portable among diagnostic systems. Such a lack of portability negates one of the major benefits for which manufacturers consider using ODX based diagnostic systems.

Another limitation of the ODX standard is its lack of definition for the communication between ECUs. The standard contains the "VehicleInfo" package, which allows for the specification of vehicle diagnostic data and ▶

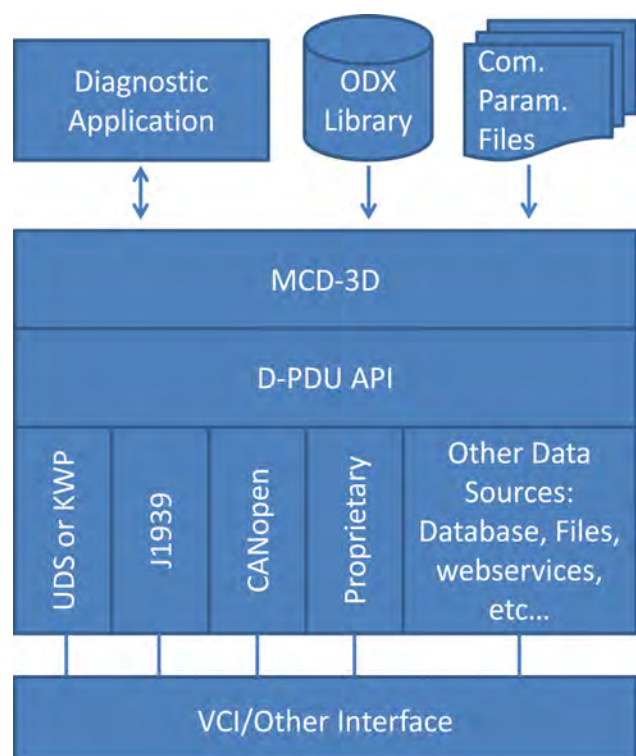


Figure 1: ODX-based diagnostic system (Photo: Sontheim)

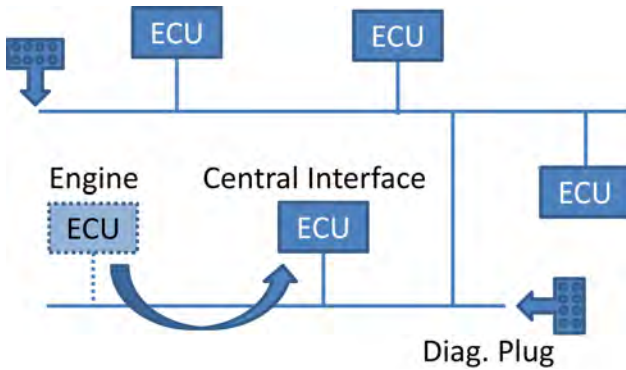


Figure 2: Sample section of a CAN network layout of an agricultural tractor (Photo: Sontheim)

its diagnostic network [2]. However, it does not describe ECUs that do not communicate diagnostic information or that do not communicate directly with the diagnostic application. Therefore, communication between ECUs in a vehicle system is not described. This lack of a description affects a diagnostic system's ability to fully define a vehicle network.

This issue is being faced by OEMs who manufacture machines containing a variety of parts from different suppliers – for example, manufacturers of agricultural tractors. Figure 2 illustrates a simplified CAN network layout which is part of a tractor's network. In this real world example, a tractor OEM uses engines from different suppliers to meet the requirements for their various types of tractors. However, the manufacturer wants to have a diagnostic tool with the same functionality, look, and feel to the service technician

for every tractor regardless of the engine it employs. To do this, each tractor is equipped with a central interface ECU, as illustrated in Figure 2. The engine ECU communicates diagnostic information directly to the central interface ECU, which then translates it from the supplier specific format into the manufacturer format. This way, the information displayed to the technician is in the same format regardless of the engine supplier. In the current standard, this type of communication is not specified. Since the engine ECU does not communicate directly with the diagnostic application it is not included in the "VehicleInfo". To get around this limitation, the diagnostic system designer has to extend the ODX format to account for such communication, creating additional work and costs and limiting portability.

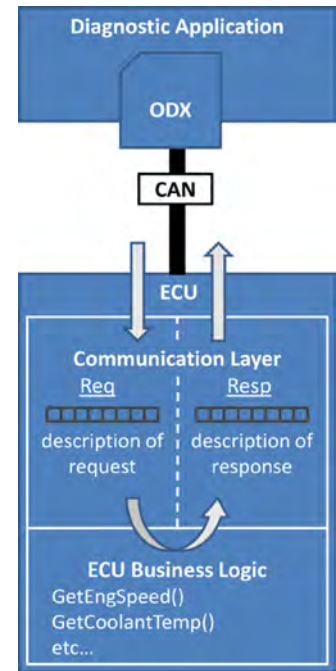


Figure 3: Communication between an ECU and a diagnostic application (Photo: Sontheim)

CAN Products for your requirements



CAN-Repeater
CRep DS 102



CAN-LWL-Router
CG FL



CAN-Repeater
CRep S8C

- Repeaters for different network topologies
- Stub line connection of networks segments
- Optical fibre connection of copper networks
- Cascadable star wiring for up to 24 CAN channels with star repeaters

EMS
Thomas Wünsche

Sonnenhang 3
D-85304 Immünster
Tel.: +49-8441-49 02 60
Fax: +49-8441-8 18 60
www.ems-wuensche.com

One additional limitation of the ODX standard is the functionality of the special data group (SDG). The standard contains an “AdminInformation” package, which allows the insertion of information used to support the diagnostics development process [2]. This information may include the name of the responsible person for the element, the name of the company, revision history, labels from a version management system, and SDGs which are used to provide company-specific data – data which would otherwise not have a standardized place within the data model [2]. However, the ODX specification limits the use of SDGs to only a few specific elements, e.g. diagnostic service elements but not parameters. As a result, diagnostic data designers include this information in other non-standard ways.

The future of ODX

The current state of the ODX standard is not yet where it must be in order to provide truly portable data that can be easily adopted by multiple OEMs in various diagnostic systems. The main reason is that it lacks definitions and functionalities, which the market has shown to be required in a complete ODX-based diagnostic system. As such, several extensions to the ODX standard are hereby proposed.

First, it is important to extend the standard to include support for more protocols. The standard should define comparams for other protocols such as CANopen or commonly used variations of standard protocols such as ISO 11783. ODX should also be extended so that it can define entire machine networks including ECUs that communicate with each other or that do not communicate directly with the diagnostic application.

Additionally, the use of ODX in industry has shown that it has the potential for more than diagnostics; it is a useful tool for describing communication in general. This is the future of ODX. As such, extensions should also be made to handle non-diagnostic and non-ECU communication. Examples include ECUs that transmit non-diagnostic information and the use of non-ECU data sources as part of a diagnostic system, e.g. files, databases, webservices, etc. Such a use of ODX is already being implemented by various companies; however, this use is not yet standardized. Finally, in addition to an extended standard for communication, the future of ODX also lies in its use as an integral part of the development cycle of ECUs in a machine system. This can be achieved through ODX-based code generation.

References

- [1] “Asam MCD-2 D V2.2.0.” ASAM Connects - Standard Detail. Web. 29 April
- [2] “Asam MCD-2 D.” ASAM WIKI. Web. 29 April 2015

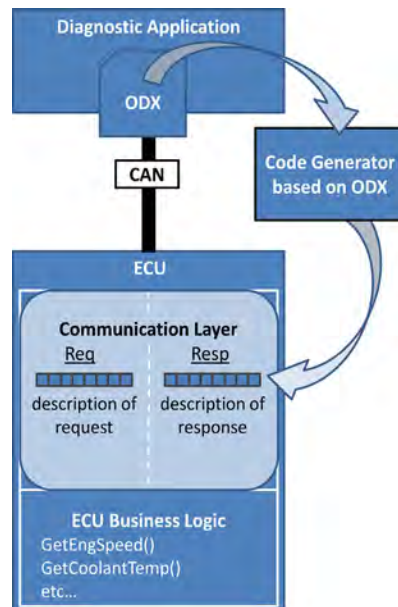


Figure 4: ODX-based code generator (Photo: Sontheim)

ODX-based code generation

Each ECU in a diagnostic network needs to make sense of requests made by the application as well as generate adequate responses. Figure 3 illustrates this basic functionality. Traditionally, ECU developers follow the same basic steps to develop ECU diagnostic firmware. First, they create the business logic, where the functions required for obtaining relevant information for diagnostics are defined; e.g. how to get the engine speed (Figure 3). Next, they define the communication layer – how to interpret received messages, how to communicate with the business logic, and how to structure outgoing messages containing ECU data. After this, the development of the ECU diagnostic firmware is considered to be finished.

For the ECU to communicate with a diagnostic application, a description of this information must be implemented in the overall diagnostic system. This is done via ODX. Unfortunately, the creation of the ODX file is often an afterthought, leading to inaccuracies and mismatches in the communication description between the ODX file and the communication layer of the ECU. Such mismatches render the ODX file unusable and additional time and effort is spent on fixing these issues to achieve a working system.

Luckily, the ODX standard describes ECU communication in a way that is complete, organized, and machine readable, making it suitable for use in more than just the diagnostic application. ODX files can also be used as part of a code generation system, which can automatically generate the communication layer of an ECU. Figure 4 illustrates this concept.

The function of the code generator is to take an ODX file describing the diagnostic communication of an ECU and then to automatically generate the communication layer for that ECU from this file. To implement a code generator, the traditional development steps for ECU diagnostic communication firmware have to change: The developers must first define and manually create the business logic of the ECU, as usual. Next, however, rather than building the communication layer, the developers describe the intended way for the ECU to communicate by creating an ODX file. Finally, this file is put into the code generator, automatically creating the communication layer software. This is then implemented on the ECU.

The goal of the code generator is to make the definition of the ECU communication via ODX a part of the development of an ECU. Changing this dynamic has several advantages: The definition of ECU diagnostic communication is no longer an afterthought since it is defined during the development cycle of the ECU – it is more likely to be accurate. Using this method also encourages the manufacturer to define ECU communication using ODX (no more spreadsheets and/or text document descriptions). Additionally, the automatic generation of the communication

layer guarantees the correct communication between the ECU and the diagnostic application. It also streamlines the ECU firmware development process, thus decreasing the time and cost of implementing the ECU. Finally, creating ODX files as a standard part of the ECU development cycle leads to an increased acceptance and use of the ODX standard in the industry, thereby strengthening it and helping improve it.

Conclusion

ODX is a very useful and powerful tool for describing ECU diagnostic communication. Many major manufacturers in the passenger and commercial vehicle industry have adopted the standard and built their diagnostic systems around it. Several major manufacturers in other industries such as off-highway vehicles have also begun to adopt the standard, lured by its potential to create modular, portable, standard-based diagnostic systems. Furthermore, industry use of ODX has shown its potential outside of diagnostics. However, the current ODX standard is not without its shortcomings. Limitations such as the lack of comparand definitions of commonly used protocols, lack of descriptions for defining ECU-to-ECU communication, and restrictions in the use of SDGs constrain the full potential functionality of ODX.

Therefore, the ODX standard should be extended past these limitations to improve its functionality. Furthermore, its use should be expanded to more than just diagnostics in order to unlock its full potential. Its use in the

industry today has shown that the future of ODX lies in its use for describing general communication in a machine system, not just diagnostic information; and in its use for generating code for ECUs.

Extending the standard, however, is not an easy task without a strong push for its adoption by the different OEMs in various industries and markets worldwide. Without such a support, it is extremely difficult to flush out its shortcomings and to strengthen it. For this reason, OEMs and component suppliers are encouraged to adopt the standard for the betterment of their own systems as well as that of the industry as a whole. ◀

Author



Juan Aguilar
 Sontheim Industrie Elektronik GmbH
www.s-i-e.de
info@s-i-e.de

CAN Products for your requirements



CTrans OL



EtherCAN CI-ARM9



CPC-USB/embedded

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



Sonnenhang 3
 D-85304 Ilmmünster
 Tel.: +49-8441-49 02 60
 Fax: +49-8441-8 18 60
www.ems-wuensche.com

Hybrid CAN and CAN FD networks

While CAN FD is quickly becoming a reality in the world of automotive in-vehicle networks, several challenges still have to be overcome. Hybrid networks of Classical CAN and CAN FD nodes could be a solution.

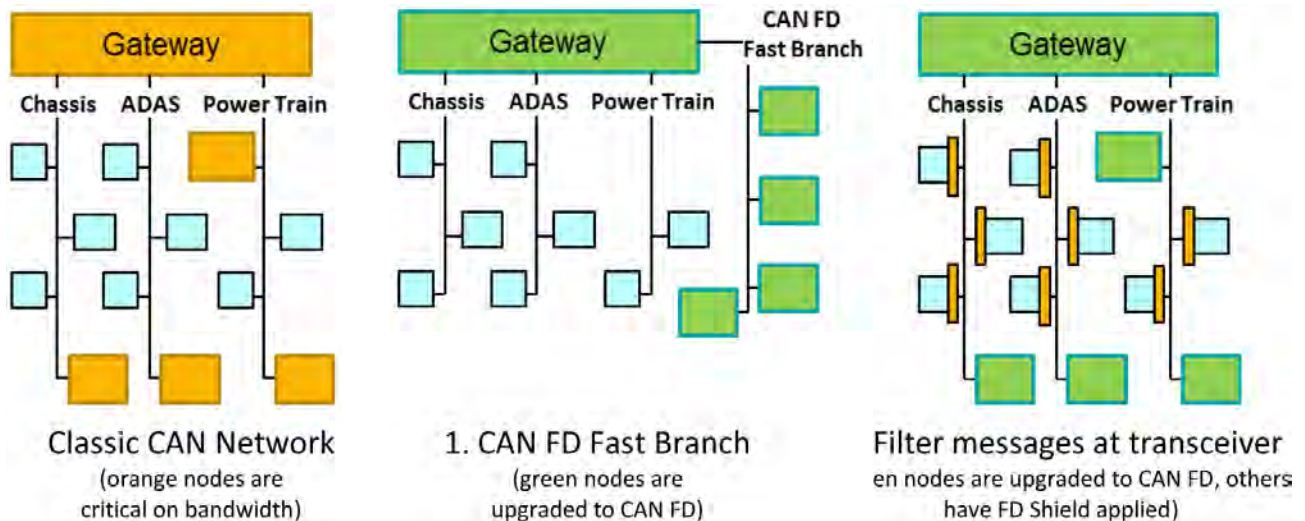


Figure 1: Options for partial CAN FD migration of a vehicle network

Many automotive manufacturers are now in full evaluation of a CAN FD introduction and over the next five years, we can expect to see these new platforms in production. This is predominantly driven by the need for bandwidth to handle more complex operations, introduce security on the CAN network and for ECU (electronic control unit) fast flashing, when software is downloaded via the CAN network onto ECUs in the production line. In fast flashing, CAN FD can increase the net-bit-rate dramatically, with a resultant reduction in flashing time. In general operation, bit-rates can also be accelerated, but are limited by EMC and network topology constraints.

Implications of CAN FD adoption

When introducing CAN FD, there are several challenges that need to be overcome, affecting both the physical layer and controllers. Firstly, new physical layer parameters need to be guaranteed supporting higher data rates of operation. These are defined in ISO 11898-2:2015, which (at the time of writing) is submitted for DIS (Draft International Standard) balloting. Many physical layer providers have already released updates to their datasheets supporting the “loop delay symmetry” parameter and subsequent updates will follow to finalize the additional parameters.

Secondly, when moving to higher data rates, the network topology needs to be verified to check all delays and ringing. To cope with this, accurate physical layer simulation models supporting data rates >1 Mbit/s are

required to ensure topologies are validated at accelerated speeds.

Lastly, and most relevant for this topic, since CAN FD is a protocol change, new CAN FD controllers are required. While CAN FD controllers can interpret and transmit both CAN FD and Classical CAN messages, Classical CAN controllers will report CAN FD messages as an error. This mandates a strict separation of CAN FD and Classical CAN networks, with every node on a CAN FD network required to support CAN FD.

The availability of CAN FD controllers is a challenge for the industry, and one currently being addressed by the industry. But even in the longer term, the necessity to make a change to bring a Classical CAN ECU into a CAN FD network remains. This will require engineering investment, a potential change in component cost (especially short term, as CAN FD controllers still emerge), and a potential module requalification, each with their own effort and cost, not to mention risk, for a network owner when transitioning from Classical CAN to CAN FD.

To minimize this impact, the most efficient approach to introduce CAN FD is to apply it only where bandwidth improvement is most valuable. Taking into account the required separation of Classical CAN and CAN FD, this essentially leaves two options: create a fast CAN FD branch through a gateway function, or upgrade all ECUs on those affected branches to CAN FD. Assuming that upgrading a complete branch presents the same challenges as a full network branch, but with fewer nodes, the discussion will be focused on the first of these options. ▶

First approach: Creating a “fast branch”

To create a fast branch is to co-locate all CAN FD nodes on a dedicated CAN FD branch, connected to other nodes via an already existing central gateway. Communication between CAN FD nodes runs at faster bit-rates and the gateway manages routing to Classical CAN nodes. This strategy is definitely preferred in networks where the number of branches is high and the number of nodes per branch is low. In this case, the transition can be quite easy and preferred in terms of operation.

For networks with fewer branches, or where the number of nodes per branch is higher, this approach can be more problematic. It implies that branches are no longer organized by function, but by technology. This creates additional routing via the gateway and prevents a domain-based security approach with rigid access control being implemented. It has also a fundamental scalability problem (if an extra ECU is upgraded, it must be moved to the physically different fast branch, on which the wiring will be non-optimized for ringing).

A different solution: hybrid networks

Having already seen that a complete upgrade of either a branch or network comes with its own costs, an alternative remains to be considered: a hybrid network of Classical CAN and CAN FD nodes, where only data intensive functions and messages are upgraded and the rest remain on Classical CAN. This minimizes upgrade costs by restricting them to only those ECUs that are required to be upgraded and maximizes the re-use of Classical CAN ECUs.

A solution to do this for the ECU fast-flashing use-case has already been realized with the introduction of the “FD Passive” extension to partial networking, available in NXP’s TJA1145/FD and UJA1168/FD. Prior to a CAN FD transmission, all Classical CAN nodes are put into selective wake-up mode with the FD Passive function active. Once completed, the CAN FD communication begins to flash the ECUs. The CAN FD Frame Identifier in the frame – the “FDF” bit – is detected in the FD passive transceiver and the frame is dropped, avoiding any CAN FD frames being seen by the Classical CAN controllers, thus avoiding any errors. Once CAN FD communication has completed, the network wakes all Classical CAN nodes and the network begins communicating with Classical CAN again.

FD Passive is an elegant solution to resolve the ECU fast-flashing use-case, but it is not applicable for general operation, due to its additional network management operations. In the ideal case, a true hybrid solution for general operation must fulfill strict requirements, in order to function correctly and deliver the true benefits of a hybrid approach:

- ◆ It must be a drop-in replacement to existing HS-CAN transceivers,
- ◆ It must not imply any software changes and must work seamlessly with Autosar,
- ◆ It must fully comply with the rules of ISO11898-1 and -2,
- ◆ It must allow CAN FD and Classical CAN messages to arbitrate against each other,

QNX and PREEMPT_RT Linux

the stable and reliable real-time platform for embedded & distributed systems

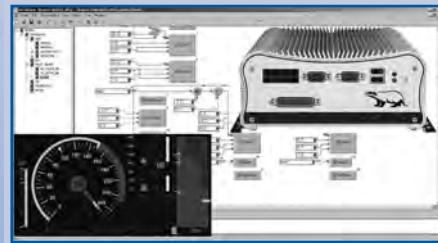
CAN | CANopen® | J1939

DACHS®

Distributed Automation Control System

Standard DACHS® products for CAN

- CAN Layer2, CANopen, and J1939 in real-time
- high performance drivers and APIs
- CANopen stack with sources
- IEC 61131-3 / IEC 61499 programming
- **DACHS**VIEW++ with C/C++ JIT Compiler



supported boards:

PC/104, PC/104-Plus, PCI, PCIe, ISA, SoCs

supported CAN controllers:

SJA 1000, i82527 or Bosch CC770, msCAN, HECC, TouCAN, etc.
for x86, PPC, ARM9, etc.

OEM solutions and adaption for OEM platforms

CONSULTING & ENGINEERING



+49 (0)64 31-52 93 66 · info@steinhoff-automation.com
www.steinhoff-automation.com · www.dachs.info

FLEXIBLE | RELIABLE | INNOVATIVE | EMBEDDED
PC-BASED | REAL-TIME | FIELDBUSES

DACHS® Product Suite, support worldwide, consulting & engineering
DACHS and the DACHS logo are registered trademarks of Steinhoff A.
All other trademarks belong to their respected owners.

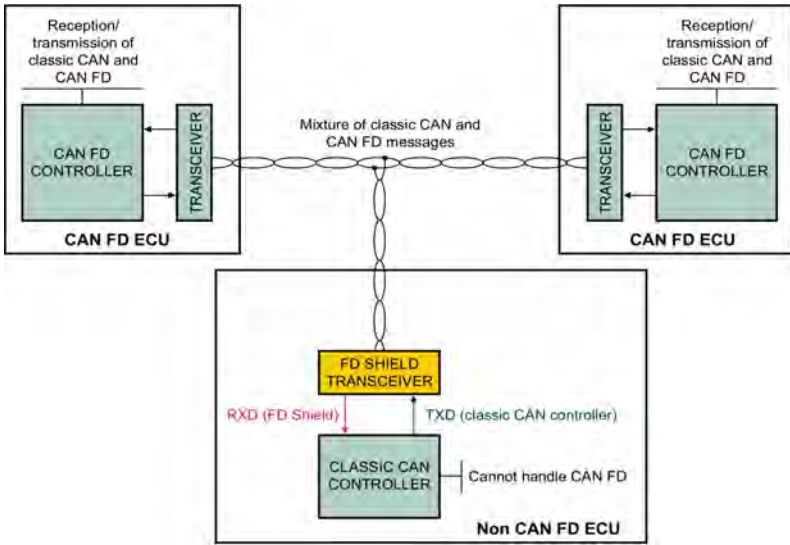


Figure 2: Technical implementation of FD Shield

- ◆ It must support all low-powers of HS-CAN transceivers (both 8- and 14-pin devices),
- ◆ It must ensure no messages are lost and all ECUs stay synchronized to the bus at all times,
- ◆ It must handle all error scenarios on the bus.

To fulfill these requirements, NXP defined the FD Shield technology – a smart transceiver able to dynamically filter CAN FD messages on the network, while being a drop-in replacement for conventional HS-CAN transceivers. No additional software changes are required, nor are any additional components; this ensures migration costs for an existing ECU are limited to changing the HS-CAN transceiver to FD Shield as a drop-in replacement.

Technical implementation of FD Shield

In its simplest terms, FD Shield essentially manipulates the TXD and RXD lines of a Classical CAN controller, based on what is received on the network. FD Shield works by having an integrated CAN FD controller and a highly accurate oscillator in the transceiver itself. As a frame arrives, the SOF and ID of the frame are passed to the CAN controller as usual. On receiving an “FDF” bit, indicating a CAN FD frame, which would cause a Classical CAN controller to generate an error, the FD Shield sets and holds its RXD output to dominant. After 6 bits, the shielded Classi-

cal CAN controller generates a stuff error, but the error frame’s TXD signal is blocked by the FD Shield towards the CAN lines, so it does not disturb the bus. The Classical CAN controller then waits for RXD to return to recessive (ISO 11898-1: “10.4.4.3 Error delimiter [...] After sending an error flag, each node shall send recessive bits and monitors the bus until it detects a recessive bit.”).

FD Shield continues to listen to the bus and at the end of the CAN FD frame (during the acknowledge field) it releases RXD to reflect the status of the bus again. This triggers the shielded CAN controller to send the (recessive) error delimiter, which concurrently occurs with the CAN FD controllers processing the end of frame field (EOF). The error delimiter and the EOF end at the same point in time, thus bringing the shielded Classical CAN and CAN FD controllers immediately back in synch; both types of controllers are now ready to start the next SOF and can arbitrate their frames against each other.

The consequence of this approach is that the Classical CAN controller increments its receive error counter by at least 9 (but possibly more) for each CAN FD frame and decrements it by 1 for each Classical CAN frame received. The Classical CAN controller will therefore likely become ‘error passive’ unless there is a high ratio of Classical CAN vs. CAN FD frames. Being ‘error passive’ means the CAN controller has to wait an additional 8 bit times after a successful transmission before starting the next (see ISO 11898: section ‘suspend transmission’). But, since the time penalty only applies to consecutive transmissions and the Classical CAN node has just lost the arbitration to the CAN FD frame, there is no additional waiting time after receiving a frame and being error passive.

As the receive and the transmit error counters are independent, there is also no risk of the shielded CAN controller entering ‘bus off’ state. A full elaboration of FD Shield’s behavior extends beyond the scope of this article, but is described in NXP’s TR1406 Technical Report >

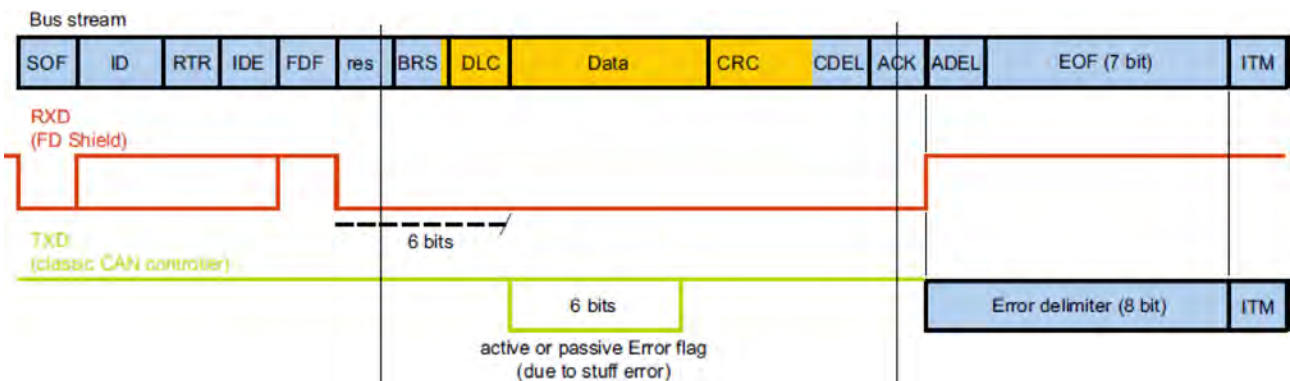


Figure 3: Time-axis view of FD Shield’s behavior when receiving a CAN FD frame: The top row shows the data on the bus; the second row shows the RXD pin of the FD Shield; the bottom row shows TXD output of the Classical CAN controller, where the error is blocked towards the bus

and covers all corner cases and implications of the error passive state.

Industry acceptance

NXP has been actively working with partners in the industry to validate this concept. The aforementioned technical performance test house and confirmed as having no blocking criteria that would prohibit its use within the vehicle. A full conformance test of the FD Shield function is also in progress at the time of writing, where the assessment is made against the official ISO "CAN FD Tolerant" test specification. Additionally, an assessment of FD Shield together with Autosar has been completed by a leading Autosar software provider, confirming that an Autosar node can handle both Classical CAN and CAN FD messages and that as the receive error counter is not passed beyond the CAN Driver interface, there is no issue with the node being error passive.

Finally, NXP is working with toolchain providers to enable automotive manufacturers to assess their existing CAN networks and understand which nodes are generating the most bandwidth, and what the effect of upgrading just these specific nodes can be on the overall network performance, to keep upgrade costs to a minimum and increase the adoption of CAN FD.

Status and plans for the future

NXP is currently developing a first silicon concept, which will have a first implementation of the FD Shield function ready for sampling in October 2015. A full product development will continue thereafter.

In conclusion, FD Shield is positioned both as an interim solution for fast CAN FD adoption while CAN FD controllers become available allowing a mix of Classical CAN and CAN FD controllers on the same bus, and as a longer term solution to avoid legacy ECU upgrade costs and maximize re-use. Unlike other strategies for the gradual introduction of CAN FD, it is fully scalable overtime, allowing additional CAN FD nodes to be ported easily without future changes to the architecture, and allows the network architecture to be function driven, rather than technology driven, with benefits for routing and easier security management. ◀

Author



Tony Adamson
NXP Semiconductors
www.nxp.com

Pioneering new technologies
Pioneering new technologies

STW[®]
30
JAHRE

Sensor-Technik Wiedemann GmbH
Mobil-Steuerungen und Messtechnik

32 bit electronic control unit ESX-3XL



- 32 bit controller with max. 136 I/Os and 4 x CAN
- Freely programmable in „C“ and CODESYS
- Certified for safety applications (SIL2, PLd)
- Including Memory Protection

Pressure transmitter with thin-film measuring element



- Pressure ranges from 0 ... 10 bar to 0 ... 2000 bar (Overall accuracy in the temperature compensated range: 1%)
- Max. media temperature 150°C / max. ambient temperature 125°C
- Wetted parts and case in stainless-steel
- CAN-Bus interface

Exhibitions



Agritechnica, Hanover
08.11. – 14.11.2015
Hall 17, Booth A34



SPS/IPC/DRIVES, Nuremberg
24.11. – 26.11.2015

Sensor-Technik Wiedemann GmbH
Am Bärenwald 6 · 87600 Kaufbeuren
Germany
Telephone +49 8341 9505-0

Controller with universal bus connections

It is difficult to find compact controllers for mobile, standalone operations, which can be integrated in diverse communication channels. Help is at hand in the form of a compact embedded PC for real-time Industrial Ethernet.

Today, modern controllers can connect entire manufacturing plants with one another and synchronize them. This permits the integration of divergent communication protocols from individual production islands or plant components. In smaller units, the range of available controllers becomes drastically narrower. If old and new communication channels converge, or if older plant components are integrated, the choice decreases further. Similarly, it is difficult to find compact controllers for mobile, standalone operations, which firstly grow with the product and, secondly, can be integrated in diverse communication channels.

Users who manufacture smaller devices or production units are confronted with the issue of the appropriate controller. It must be compact and yet upgradable to permit further developments. The next question is then: How will the controller be programmed? While in medical engineering, for instance, the application is often programmed in C for historical reasons, in other areas the use of soft PLC solutions according to IEC 61131-3 is becoming more and more prevalent. However, both worlds require compact, cost-effective solutions for a controller that can be integrated in a range of existing networks. HMS Industrial Networks has taken on this target group and has developed a solution for a compact controller for real-time Ethernet and classical fieldbus systems with the Ixxat Econ 100.

Flexible handling

Production or packaging with handling robots has become indispensable. But a small handling device for flexible use on-site has completely different requirements on the controller than large equipment. Of course, dimension and weight do play a role, but what is more important is the computer's ability to last the course when operated in different locations. If an electrical motor is running nearby, or the charging station for a forklift truck, this can result in voltage drops. In traditional controllers, this problem is solved by a cost- and maintenance-intensive UPS, which adds weight and takes up space. Use of the continuous operation-capable Econ 100 provides a solution in this case. The integrated, non-volatile memory retains all relevant data on the gripper load, the position of the boom, and other sensor and motor-specific data in the event of power failures. If the power is resupplied a few seconds later, operation is continued seamlessly.



Figure 1: Ixxat Econ 100 – Flexible controller solution for a variety of applications (Photo: HMS / © Alterfalter Fotolia.com / © zlikovec Fotolia.com)

er is resupplied a few seconds later, operation is continued seamlessly.

Apart from this “freeze and go” function, it is important for the user to be able to communicate within different networks. Since the mobile device is deployed in facilities with highly diverse infrastructures, integration in a range of communication networks is important. The third point was future viability. Even if the device currently works with a gripper, further variants with a bucket or fork are being planned. This means that the additional sensors and drives required for this purpose must be easy to connect. The compact data center, with its numerous local I/Os and communication interfaces, also stands out here. Additionally, many existing fieldbus systems can be integrated via the Econ 100 HMS Anybus module interface.

Putting on pressure

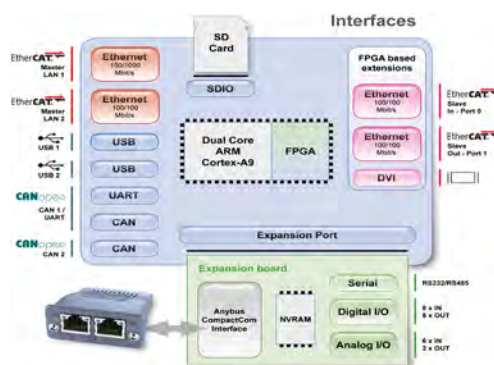


Figure 2: Interfaces and functional modules on the Ixxat Econ 100 (Photo: HMS)

Control of a special printer for safety labels is a further example. In a printer, the printing mechanism, the label stock, and the print-out take up the limited available space, which means there is little room left for the controller. Furthermore, any customer should be able to connect and operate such a printer in their communications network. Since the Econ 100 is capable of combining all bus systems via Anybus Compactcom, it is

the ideal datacenter in this case. The controller's computing power is even sufficient for the highest print resolution at high speed and is therefore comparably cost-efficient, since the system can be adapted to meet requirements using expansion cards. Additionally, the manufacturer was able to connect all previously available print modules still based on EIA-232 and EIA-485 directly to the serial ports available in the Econ 100. This makes it possible to use this controller in multi-generation systems, which combine serial communication with fieldbus systems and real-time Ethernet.

Last but not least, the range of options offered by the controller may also be of interest to OEMs. As a platform for proprietary special formats, customer-specific requirements for shape, color, connector enablement, software and hardware, and the required interfaces can be tailored to the requirements profile.

Detailed technical specifications

As an ARM-based embedded PC platform for the top-hat rail, the controller measures 72 mm x 154 mm x 105 mm. With its Linux operating system, it offers multi-protocol support as an embedded PC. In this way, customer-specific gateway and controller solutions can be implemented for a variety of different fieldbus and Industrial Ethernet standards. A high-performance CPU, up to 256 MiB RAM, a fan-less design, and an increased temperature range of -40 °C to +70 °C make it flexible for a variety of applications. Applications with critical voltage supply in which the last operating state, including all process variables, must be saved are covered by the NVRAM option.

The interfaces – two CAN, two Ethernet, and two USB – come as standard and can be expanded by a variety of interfaces using an expansion board. Analog and digital I/Os, serial interfaces, 512 KiB NVRAM, and a slot for the Compactcom modules are available. The Compactcom modules cover common fieldbus and Industrial Ethernet protocols, and can be integrated in the application software. On top of this, the expansion card has 24 inputs and outputs, e.g. for directly connecting sensors and actuators. The digital outputs, providing up to 2 A of output current and 12-bit resolution analog channels, permit a range of application-specific options when selecting the components to be connected. Two additional EIA-232/-485 interfaces on the expansion card provide a link between real-time-capable Industrial Ethernet and CAN-based networks or serial applications. In addition to programming in C/C++, HMS offers a Soft-PLC programming environment consistent with IEC 61131-3 for programming and configuring control applications in conjunction with a soft PLC manufacturer. The software package supports protocols including CANopen, Ethercat, Powerlink, Profinet and Ethernet/IP.

Author



Thomas Waggerhauser
lxxat/HMS
www.ixxat.de
info@hms.se

Temposonics®

Magnetostrictive Linear Position Sensors



In line with your application requirements MTS Sensors delivers sensor solutions which fit your needs in term of design and performance. Temposonics® position sensors with CAN-interface are the first choice in factory automation, fluid power, plastic processing, material handling and mobile hydraulics.



MTS Sensor Technologie GmbH & Co. KG
Tel. +49 (0) 23 51/95 87-0 • www.mtssensors.com

From concept model to production code

In the industry, model-based design is utilized more and more often. Because changing a conventional development process to a model-based process is not simple, we take a look at a practical approach.

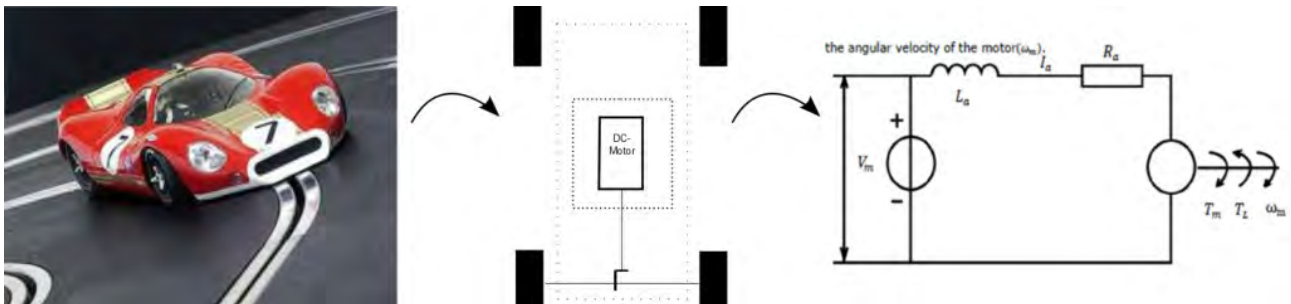


Figure 1: Slot car representing a linear actuator system and schematic model of a DC motor

The model based design (MBD) process relies on mathematical models and simulation. Adjusting to this new process can be overwhelming when not properly implemented. We present a practical approach to the MBD process, using an example to explain the critical choices that need to be made in order to start with a concept and end up with production code.

We want to illustrate the steps that are needed to go from concept model to production code in a quick development process. The process is applied to a linear actuator based on an electric DC motor. A linear actuator is used in many applications, for example agricultural tools, printers, CD players, etc. For simplification purposes, in this article the linear actuator is replaced by a slot car. Of course, the presented techniques are valid for a large range of control problems. The idea of the slot-car example is as follows: A slot car racetrack is modified so that one car is controlled by a computer. The goal is that the slot car is accelerated from a starting position as fast as possible and crosses a finish line. However, it should brake as well, because of a fictive wall shortly after the finish line. Controllable parameters are the voltage and current supply to the slot car. Position sensors for feedback are also employed.

The challenge in this example is the development of the controller. It is a perfect example to show the model based design process for a quick translation from concept model to production code. In the example, we will make use of a plant model for the development of a control design solving the control problem. The developed controller is discretized, CAN-communication is added, other software limitations are taken into account, and fault behavior is introduced. All these steps have their limitations and affect the performance of the control software on the production ECU. The plant model is used for testing and verifying the performance at each development step. This helps detecting limitations and hick-ups early in the development process. Solving these issues as early as possible in the

development process is the key to a quick and cost-effective translation from concept to production code.

Plant model

Before the development the actual controller, a plant model has to be made. A plant model helps the development process in several ways: By making a plant model, the developer gets a lot of insights in the system and how it behaves. This helps to focus on the actual control problem. Furthermore, a plant model is useful to do quick 'verification' iterations during the controller development. Creating a plant model requires a significant effort early in the project. However, it supports the development steps of the control system and helps getting closer to a first-time-right concept.

A plant model can be made in several ways, from very simple to extremely complex. Creating an appropriate plant model is a study on its own. One can easily loose oneself in making models too detailed. It is best to keep the model as simple as possible. A simple model is easy to maintain and already gives a lot of insight. If really needed, the model can be extended with additional complexity. In the slot car example, this means that we have to create a plant model of the slot car itself. It does not make sense to model the slot car as a multi-body dynamic model, which takes friction losses of the air drag, slip of the wheels, thermodynamics of the motor etc. into account. It has to be kept as simple as possible. If a linear actuator were modeled, the plant model would also not take every possible influence factor into account.

We start with a model of a DC motor, the heart of the slot car (see Figure 2). If it turns out that the dynamics of the DC motor are not sufficient to cover the behavior of the real plant, one can decide to extend the model with, for example, a load estimator (mainly friction in driveline). The DC motor is modeled according to the first-principles by

using the differential equations in the electrical and mechanical domain.

$U_m - R_a I_a - L_a \frac{dI_a}{dt} = U_{emf}$, with U , R , I , and L voltage, resistance, current, and inductance. Indices m , a , and emf represent the motor, armature, and electromotive forces.

$T_m - T_L = J \frac{d\omega_m}{dt} + D_m \omega_m$, with T , J , D and ω as torque, inertia, dynamic friction, and rotational speed. Indices m and L represent motor and load.

The coupling between both equations is given by $U_{emf} = K \omega_m$ and $T_m = K I_a$, with K as a motor constant.

When these equations are rewritten and transferred to the frequency domain, it results in the transfer function:

$$tf(s) = \frac{\omega_m(s)}{U_m(s)} = \frac{\frac{K}{L_a J_m}}{s^2 + \frac{(D_m L_a + J_m R_a)}{L_a J_m} s + \frac{(K^2 + R_a D_m)}{L_a J_m}}$$

Of course it is of great importance to use the correct parameters in the equation to have a system response which reflects the real plant. Some parameters can be measured, derived or are given by the supplier. A way to get the unknown parameters is to do verification measurements on the plant and derive these parameters from the real plant response. This is also a check if the plant behavior is sufficiently covered by the model.

Controller design

Once a plant model is created, it can be used efficiently for the controller development (Figure 3). Classical control theory is used to come up with a controller, which fulfills the set requirements. By using the plant model, the performance is easily verified and visualized, as can be seen in Figure 4.

Theoretically, the developed controller should perform very well and in most organizations, this is the end of the R&D process. Since – in theory – it has been proven that the system works, the concept is given to the software developers with the request to “Please implement this in an embedded system”. However, in the next development steps, when it has to actually be implemented on an ECU controller, performance might, and in most situations, will be affected.

Discrete controller

Once it has been proven in theory that the concept works, it must be further developed to production code. One of the necessary steps is to discretize the controller. In the end, ECUs are not able to run continuous calculations and they have limitations. By discretizing the controller to a certain sample rate, the behavior of the system is changed (see Figure 5). It is therefore of great importance to recheck the performance of the controller after discretization.

In case of the slot car, the open loop transfer function in the continuous domain looks like this:

$$G(s) = \frac{V_{vehicle}(s)}{U_m(s)} = \frac{A_c}{s^2 + B_c s + C_c}$$



Absolute Rotary Encoders and Inclinometers

Reliable Measurement under Harsh Conditions

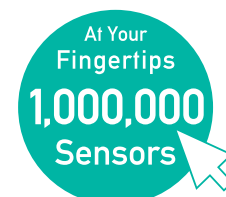
High Protection Class: IP69K

Fieldbus and Analog Interfaces

Safety and ATEX

Ex-Proof Versions Available

Successfully Integrated in
Concrete Pumps, Drilling Machines,
Working Platforms, Cranes, Wheel Loaders,
Leader Masts and More



www.posital.com

After discretizing it to a certain sample rate, the transfer function changes to:

$$tf(G) = \frac{A_d Z + B_d}{Z^2 + C_d Z + D_d}$$

Note that the values $A_d \dots D_d$ differ when the sample rate changes.

If the discretized controller is used without rechecking the performance, stability, and robustness, there is a serious risk of malfunction. With the discretizing, an additional delay is introduced, which especially affects the phase of the system. In the continuous domain, the controller is tuned with a bandwidth around 200 Hz, where the classical control theory stability margins are fine. If the discretized controller of 1 ms is used, one can see in Figure 6 that the phase margin is critically low, resulting in an unstable system. For a 10 ms discretized controller it is not even possible to reach the bandwidth of 200 Hz. To ensure correct behavior, the controller must be re-tuned. Mostly, the bandwidth has to drop significantly to guarantee a stable system. The example indicates the importance of taking these limitations into account during the development process. This avoids that R&D comes up with a controller bandwidth that is out of reach of the embedded system.

Software code must have protections against division by zero, must make decisions on signed versus unsigned variables, data types based on the possible range of a variable, etc. And secondly, solutions that could potentially work might have a significant impact on processor load and memory usage.

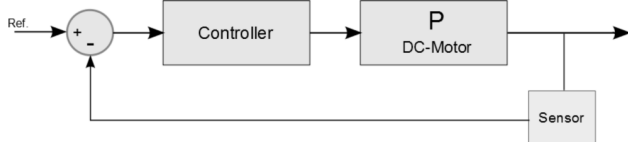


Figure 2: Controller and plant

CAN controller

The steps mentioned above still leave out an important factor: the CAN network. In typical automotive control systems, multiple control units hold sections of the complete system. In our example, a smart sensor/actuator system handles all the measurements and power electronics while the control algorithm itself runs on a separate ECU. In between is the CAN network. The CAN network has its own message rate, which acts as another discrete sample rate. However, the difficulty with CAN messages is that the actual sample rate is not constant. CAN messages are sent based on priority when there is room available on the bus. Depending on the priority of the message and bus load, additional delays may occur. The control system must be prepared to work with a bandwidth of CAN delays. Therefore, some tolerance must be engineered in for controller stability and for controller settings to meet the requirements.

Another potential delay in the process is the sensor signal processing. However, this delay is stable and less significant. Taking these types of issues into account early

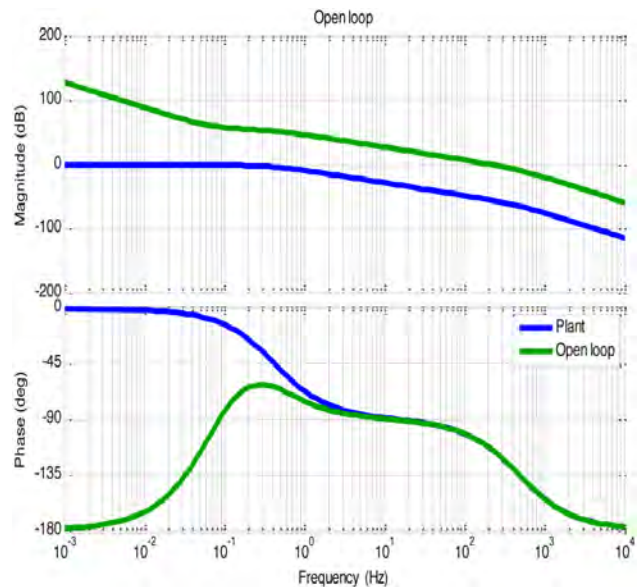


Figure 3: Bode plot of the plant and open loop controlled plant

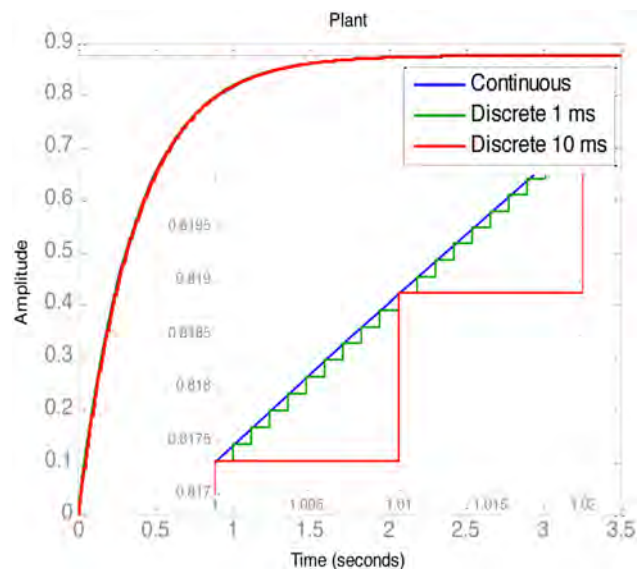


Figure 4: Effect of discretization

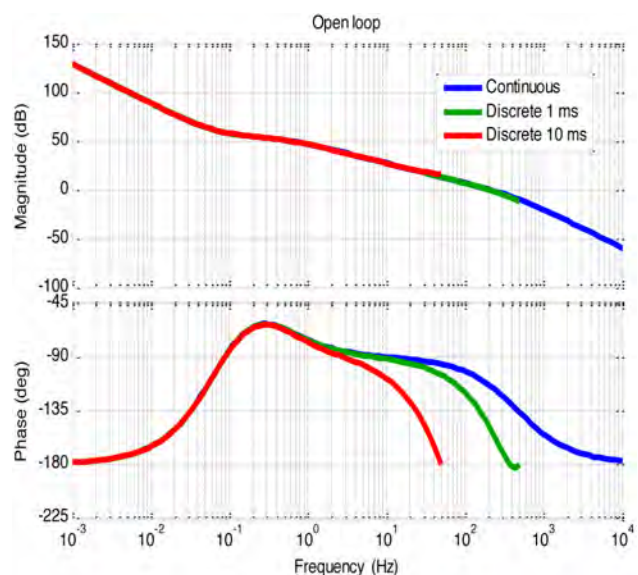


Figure 5: Bode plot of the controlled system in continuous and discrete sampling steps

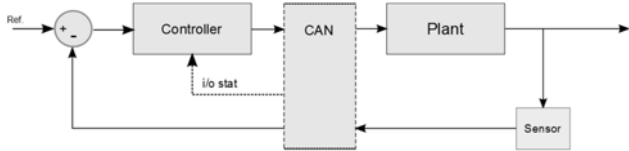


Figure 6: Controller and plant with CAN network interaction

in the development process when the controller settings are determined, limits time consuming rework during the project.

Fault behavior

A proper controller design takes into account what to do when errors occur. In the example with the CAN network interaction, it must be prevented that extreme delays of CAN messages cause unstable behavior.

For a linear actuator that has a smart sensor/actuator setup, a problem can occur when the actuator command is sent out but not quickly followed by a new actuator command. It is possible that the linear actuator is sent to its end position at maximum speed and does not slow down or stop when it reaches the end position. When trying to act on sensor signals, if the sensor CAN message has a very large delay it might not be received at all. And naturally, the sensor signal may have an error in it as well.

To prevent unwanted scenarios and damage, robustness for faulty behavior must be implemented in the controller design. Preferably, the controller enters a special safe state when faults are detected. In automotive applications, this state is typically known as “limp-home” or “default” state.

Results

The chosen approach to the linear actuator/the slot-car, provides a pragmatic approach that is time and cost effective. Quick development iterations are available to the concept developers and work can start using the plant model when the actual plant is not even available yet. The same is true for when the actual ECU target is not available yet.

The chosen approach with different pragmatic testing steps naturally guides developers to gradually build up the realism of testing scenarios and equipment. Step by step, the developers are faced with real life restrictions. This helps to make the control system more suited for its real-life environment. Involvement of the original concept developer in this process makes sure that the optimum result is achieved given the limitations of discretization, CAN networks, and other items. Fault behavior and actions to be taken towards a safe response are best identified early in the process. The developer of the plant model is also most aware of system responses and the possible risks of incorrect controller behavior.

Avoiding the pitfalls of the MBD approach

The plant model approach takes more effort at the start of the project. However, it provides many efficiency gains later on in the development process. Classis control theory

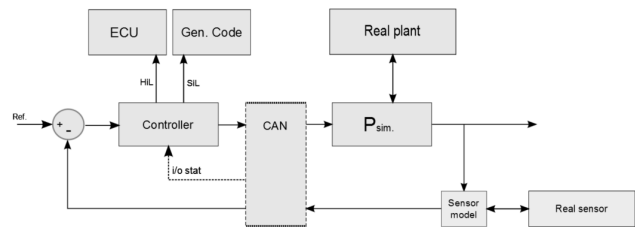


Figure 7: Controller and plant can be exchanged in development steps with simulations, software code, and real components

is still very usable for the concept phase and can be combined with the plant model. There are, however, significant pitfalls in the process.

If departments isolate themselves from each other, the benefits of the MBD approach cannot be achieved. The pitfalls of discretization, CAN network delays, and variation etc. must be tackled in cooperation between departments. Pitfalls can be identified more easily when tests and validation steps occur as early as possible. During the further course of the project, the tests involve more variables and unknowns, more actual components and targets and this gradually brings the control system to a production-ready status. Handling the testing and verification/validation steps late in the process mostly results in a large amount of rework, which is costly and time consuming. ◀

Authors

Mark Maessen, Judith Vliegen
Brace Automotive
www.brace-automotive.com

Squirreling away solar energy for winter

An innovative storage power plant stores the energy produced by photovoltaic systems as hydrogen. By employing the system, smaller companies can reduce their carbon footprint to zero.

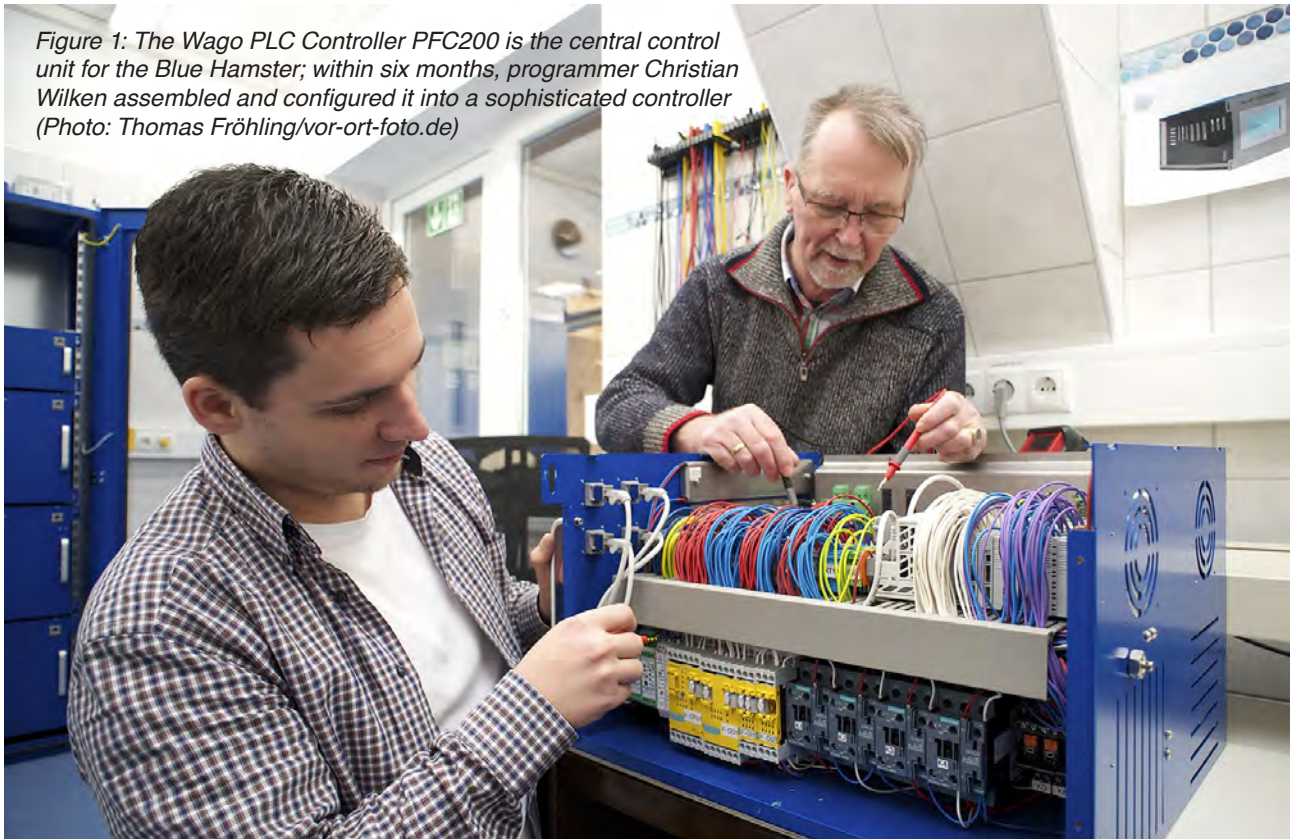


Figure 1: The Wago PLC Controller PFC200 is the central control unit for the Blue Hamster; within six months, programmer Christian Wilken assembled and configured it into a sophisticated controller (Photo: Thomas Fröhling/vor-ort-foto.de)

The stereotype paints the people from Germany's far north as moving at a more leisurely, lazier pace. When applied to Mossau Energy, a specialist in photovoltaic systems headquartered in the East Frisian city of Aurich, this cliché falls rather flat. "We've always been pioneers," states Helmut Janßen, Dipl.-Ing. The company has only 19 employees, yet they enjoy the luxury of an in-house research and development department. This has led to enormous advances in their systems. "We have developed the first marketable system that can store renewable energy for long periods of time, the Blue Hamster," declares Janßen, who is in charge of its development. Internal control of the system is directed by an I/O controller from the Wago-I/O-System 750.

Using solar energy in winter

Perhaps it's due to the unobstructed vistas that the East Frisians enjoy. In a place where dykes are the only thing obstructing a view of the ocean, ideas for new energy technology have found especially fertile ground. Company

founder, Günter Mossau, brought photovoltaics to East Frisia in the mid-90s. According to Janßen, people sneered at the concept then. After all, the north is not generally known for an excess of sunlight; wind energy is the more obvious choice. Yet, wind is also important for photovoltaic systems, since the better one is able to cool the temperature-sensitive collectors, the higher the electrical yield. "You can recognize Mossau roofs from far away," confirms Janßen. While other photovoltaic companies exploit every square centimeter on the roof for their collectors, the Mossau installers always leave gaps between the individual modules. This allows the East Frisian winds to pass between and around the arrays, cooling them in the process. "Mossau Roofs" yield around ten percent more electricity from the sun than comparable installations.

However, it continued to annoy the detail-oriented Günter Mossau that the increases in efficiency were lost, fading away with the setting sun. The technologies for storing the solar energy he could produce lagged far behind his needs. At peak times, the photovoltaic arrays and wind farms located in Germany generate more energy than can

be consumed. Yet storage facilities for holding this massive amount of electricity as a supply against slack times are basically non-existent. Which means that as soon as the sun sets and the wind dies, conventional power plants have to take over. On the road to renewable energy sources that are clean and independent from fossil fuels, the storage of electricity from volatile sources, like solar and wind, remains a core problem. "Our idea was to solve the storage problem for home owners and small businesses," explains Janßen. "Our systems are supposed to store the electricity generated by the photovoltaic system in the summer, when it isn't immediately needed, in order to make it available in the darker days of winter."

Splitting water into oxygen and hydrogen

In order to achieve this, the systems designed by Mossau Energy rely on electrolysis, a method that is well-suited for storing the large amounts of electricity generated by wind and solar for longer periods of time. In this process, electricity is used to split water into its constituent parts, oxygen and hydrogen. The oxygen is released to the environment, while the hydrogen is stored in a tank. Experts call this "Power to Gas". In the winter, the electrolysis is reversed using fuel cells; the hydrogen and oxygen are fed back together and reacted. This generates electricity with water as a by-product.

Electrolyzers and fuel cells are hardly new. However, combining them into a marketable product that makes renewable energy available year round? That was novel.

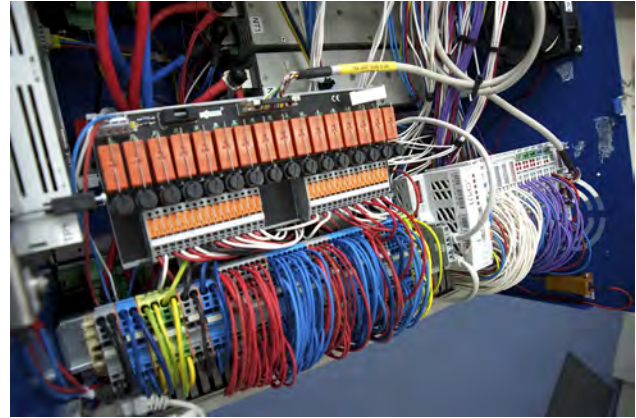


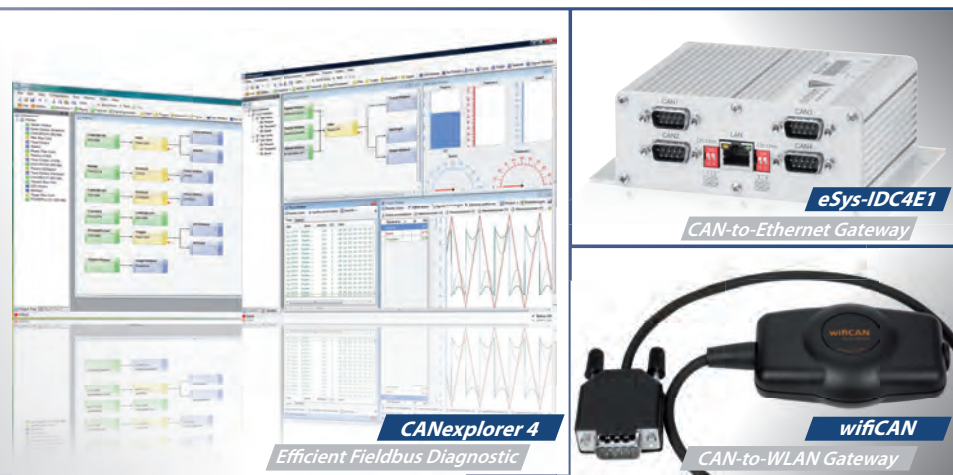
Figure 2: The Blue Hamster converts current from a photovoltaic system into hydrogen; a fuel cell reverses the process as needed - the processes are controlled by the Wago-I/O-System 750 (Photo: Thomas Fröhling/vor-ort-foto.de)

It took the company three years before they could present their first prototype in 2013. Mossau Energy promptly received the "Federal award for outstanding innovation in industry and trade" from the Federal Ministry for Economics and Labor for their Blue Hamster. One year later, they had developed the prototype into a compact, market-ready system.

It is hard to imagine from the outside that the blue control cabinet, around the size of an average adult male, houses this innovative, small storage power plant. Janßen elaborates, "The system stores solar energy according to ▶

Wir leben Elektronik!
We live electronics!

sontheim
Industrie Elektronik GmbH



- ▶ Mobile or stationary CAN Interfaces in various form factors with WLAN, Ethernet, USB and more
- ▶ Robust and flexible CAN Gateways and Data Logger with up to 256GB of built-in NAND-Flash-Memory
- ▶ Rugged ECUs for controlling, telemetry services and diagnostic application
- ▶ Monitoring and analyzing - our modular software tools for an efficient fieldbus diagnostics
- ▶ Searching for a modular diagnostic tool based on standards?

<http://www.s-i-e.de/en/products/diagnostics/mdt>



Agritechnica 2015, Hanover
08.11 - 14.11.2015
Hall 15, Booth F49

Sontheim Overview and Portfolio:



Start to Get Free - More Flexibility With Our High-Performant CAN Solutions

As your reliable partner for innovative CAN systems we support you with our tools in all phases of your projects, from design over implementation to testing.

We live electronics!
www.sontheim-industrie-elektronik.de

DE Sontheim Industrie Elektronik GmbH
Georg-Krug-Str. 2, 87437 Kempten
Tel: +49 831 57 59 00-0 - Fax: -73
info@s-i-e.de

US Sontheim Industrial Electronics Inc.
One West Court Square, Suite 750
Decatur, GA 30030
Phone: +1 (404) 494-7839 - Fax: -7701



Figure 3: Fuel from the hydrogen tank: The Blue Hamster can completely supply a company, whose electrical consumption is less than 100 000 kW hours per year, with clean, solar energy; that which is squirreled away in the summer can be used during the winter (Photo: Thomas Fröhling/vor-ort-foto.de)

various priorities.” Initially, the current consumption needs are satisfied. If more energy is produced than consumed, then the Blue Hamster fills a short-term buffer based on lithium-ion batteries. These are already sufficient to bridge one to two sunless days. Once the batteries are fully charged, the Blue Hamster converts the additional excess energy into hydrogen. By storing this hydrogen during the sunny months of summer, it is available for a fuel cell to use later. This allows the system owner to “squirrel away” enough green energy for the cold days of winter, just like a hamster stores food for later.

Brains from Minden

The functional principle of the Blue Hamster may sound simple, but it is actually a very complex issue. At every point in time, the system must know how much solar energy is being produced, how much of that energy is being directly consumed or supplied to the national grid, the charging status of the batteries, and the fill level of the hydrogen tank. In addition, the chemical processes

in the electrolyzer and in the fuel cell must be constantly monitored. In short, the Blue Hamster needs a brain. Janßen reports, “We initially were considering a controller from another manufacturer. However, we decided to use products from Wago, because they measure up to the challenges presented by our systems.”

The central control unit for the Blue Hamster is the 750-880 PLC Controller. Additional Wago modules include a 3-Phase Power Measurement Module (750-494), a 2-Channel Analog Input Module (750-461), a 4-Channel Analog Output Module (750-559), and multiple 4-Channel Analog Input Modules (750-455) to record the data and control the different currents. In addition, all data are stored on an SD card; the slot for the card is an integral component of the PLC controller. The fuel cell can be linked to the Blue Hamster main controller using a CAN network. All data, which are important for monitoring the entire system, are exchanged using the CAN network, including fuel cell data, output currents, voltage, temperature, operating hours, and the amount of energy that is fed into the grid.

According to Christian Wilken, who is responsible for programming the Blue Hamster, even though the fuel cell doesn’t need to be controlled, it is highly advantageous that communication in both directions is possible through the CAN network. Since the CAN network can be used to write data as well as read it, it is also possible to specify some values for the fuel cell as well as to read the data it produces. These can be, for example, performance targets or current limits. “Since the processes that run in the fuel cell are already quite complex, it is good that the CAN bus provides a standardized external interface,” explains Wilken. “This provides numerous opportunities to visualize data and react at the correct time if the parameters develop differently than they should.” In addition, the CAN interfaces offer the possibility of integrating the additional battery storage, including the battery management system, into the Blue Hamster. Previously, this system had been linked in externally. This is therefore the newest idea that Mossau Energy intends to pursue as they further develop the Blue Hamster. ▶

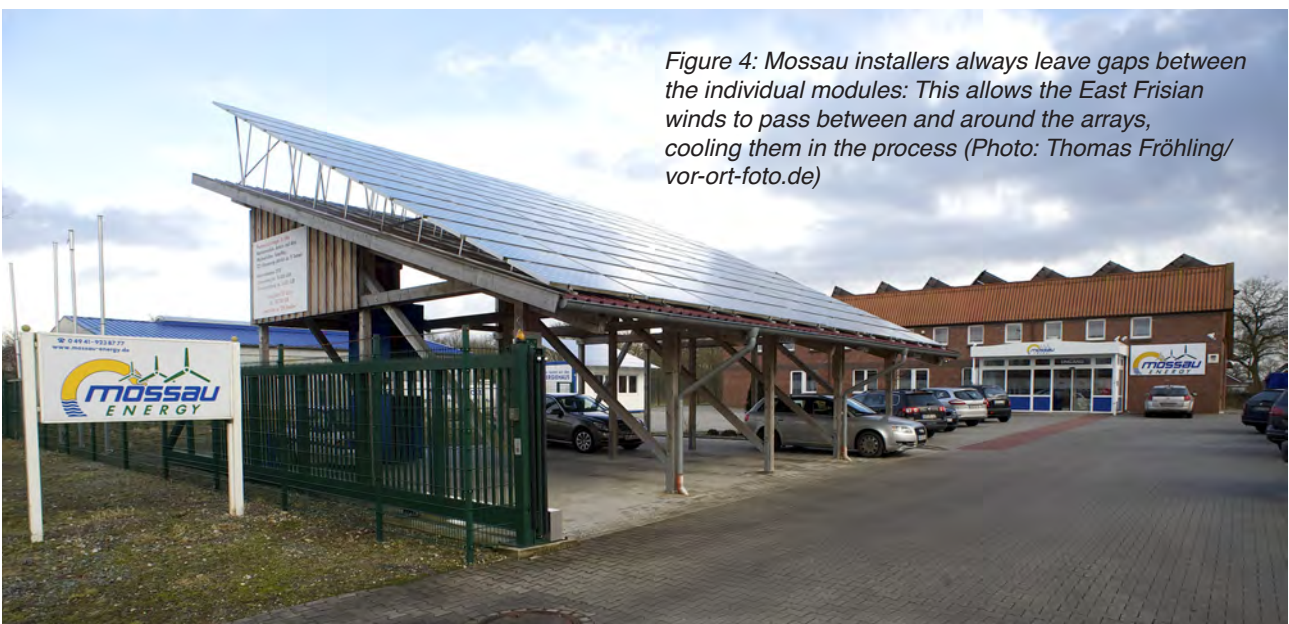


Figure 4: Mossau installers always leave gaps between the individual modules: This allows the East Frisian winds to pass between and around the arrays, cooling them in the process (Photo: Thomas Fröhling/vor-ort-foto.de)

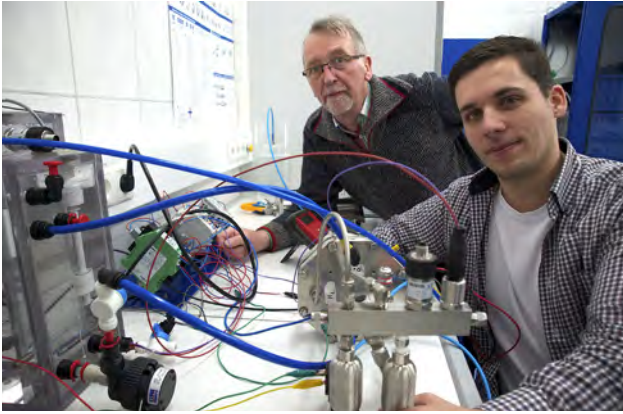


Figure 5: The engineers at Mossau Energy developed the Blue Hamster in their own laboratory – quite a feat, for a company with only 19 employees (Photo: Thomas Fröhling/vor-ort-foto.de)

A touchscreen from Wago is incorporated into the door of the control cabinet, which can be used to call up current system information. “It took about six months to set up the entire system,” reports Wilken. Although Wilken had never worked in the Codesys programming environment, a two-day introduction was sufficient to allow him to set up the Blue Hamster. “Whenever I had any questions, I could immediately get them clarified by my contact at Wago,” reports Wilken.

First reference projects realized

A single Blue Hamster can cover annual consumption of between 25000 kW and 50000 kW hours. For higher consumers, the systems can be duplicated. Mossau Energy has been supplying their own needs with a Blue Hamster for quite some time. Another installation is located at Klar Folien, headquartered in Dernbach in Westerwald, a village in the Rhineland-Palatinate. There are currently plans to establish a global marketing system in order to increase use.

Due to the very expensive components – the electrolyzer and the fuel cells, which are not yet manufactured in series production, and also the hydrogen tank – the Blue Hamster is an expensive piece of equipment. In spite of this, a target group can already profit from the Blue Hamster: “Small companies can reduce their carbon footprint to zero, completely divorce themselves from fossil fuels supplied via the electrical grid, solve an image problem, or provide proof of sustainability in their practices,” elaborates Janßen. Ideas from East Frisia make it all possible.



Author

Heiko Tautor
Wago Kontakttechnik GmbH & Co. KG
www.wago.com

1985

30 years ago, it was all about establishing communication ...



The same year, 1985, Kvaser was founded by a far-sighted group of engineers to exploit the future potential of CAN for electronic control systems.

Find out more by visiting:

www.kvaser.com/about-us/history



30 YEARS OF FORESIGHT



Find CAN hardware and software at
www.kvaser.com

YOUR PORTAL TO THE PERFECT CAN SOLUTION

CAN FD: from theory to practice

The international standardization of CAN FD is settled. The next step is the development of recommendations and specifications, how to design CAN FD networks.

The CAN FD data link layer submitted to ISO, the international organization for standardization, has passed the DIS (Draft International Standard) balloting without negative votes. This means, after implementing the observed comments, the ISO 11898-1 standard will be published. More than 100 comments, mainly of editorial nature, are already observed and implemented. Now, it is just a matter of time, when the ISO 11898-1 document will be published as International Standard. This standard also specifies a part of the physical layer, the physical coding sub-layer, according to the OSI (open system interconnection) reference model. The CAN FD physical media attachment (PMA) sub-layer describing the transceiver characteristics is internationally standardized in ISO 11898-2. This document has been submitted for DIS balloting. It comprises also the optional low-power mode (formerly in ISO 11898-5) and the optional selective wake-up functionality (formerly in ISO 11898-6). System-related specifications have been deleted. The new ISO 11898-2 standard specifies just the transceiver characteristics. The physical media dependent sub-layer is not in the scope of ISO 11898 series. It is highly application-specific, and might be specified by other ISO standards or other associations (e.g. CiA, IEC, or SAE).

ISO standardizes also the conformance test plans for ISO 11898-1 and ISO 11898-2 implementations. The related standards, ISO 16845-1 respectively ISO 16845-2 are under development. ISO 16845-1 has been submitted for DIS balloting and ISO 16845-2 is in CD (committee draft) voting. The test houses C & S Group and IHR are already developing conformance test prototypes. Most of the automotive chipmakers are in the process to integrate CAN FD cores into their micro-controllers. Many of them have licensed the IP module from Bosch. There are also several vendors providing a self-implemented CAN FD ASIC/FPGA. Engineering samples of CAN high-speed transceivers qualified for 2 Mbit/s and 5 Mbit/s are already available by several companies.

Next step: using the longer data frames

The standardization of the CAN FD data link layer and physical layer is settled. Even the first higher-layer protocols (e.g. transport layer and application layer) make use of the longer CAN FD data frames. This includes the so-called ISO transport layer as standardized in ISO 17765-2. First implementations by Vector and Volkswagen have been tested last October during a CAN FD plugfest organized by CAN in Automation (CiA). Also the XCP calibra-

tion protocol version 1.2 specified by the nonprofit ASAM association makes use of the 64-byte data fields. Autosar version 4.2.1 supports also CAN FD.

The CiA CANopen SIG (special interest group) application layer develops currently the CiA 301 version 5.0, which will be based on the CAN FD data link layer. Most of the CANopen protocols will remain as they are since more than 20 years. However, the PDOs will be prolonged to 64 byte. The number of mapping entries will be still 64. This means, it is not possible to map more than 8 byte bit-wise. Byte-wise mapping is preferred. The bigger change regards the SDO protocol. It will be completely changed: The Universal SDO protocol will be structured better and will be easier to implement. It is still under development.

The SAE is discussing in its J1939 CAN FD task force, how to make use of the longer frames. In parallel, the CiA IG (interest group) commercial vehicles prepares the CiA 602-2 application layer proposing a mapping of J1939-71

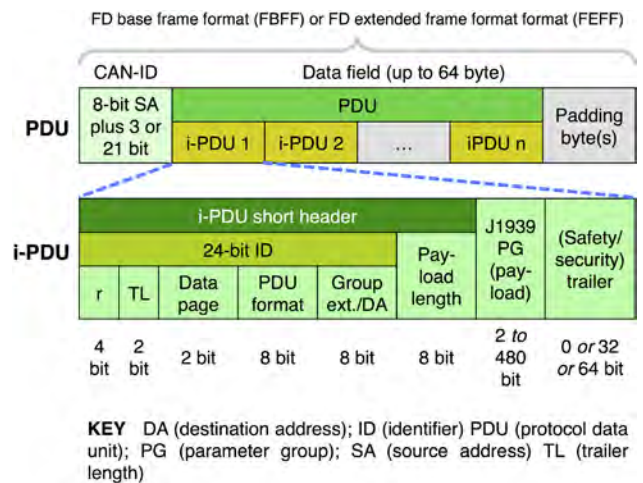


Figure 1: The proposed protocol structure for commercial vehicles complies with Autosar (Source CiA 602-2)

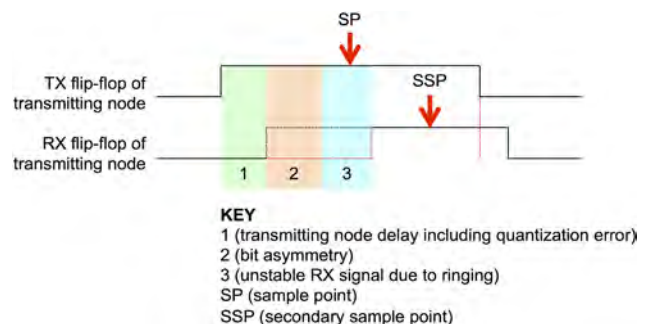


Figure 2: Sampling of a recessive bit at the transmitting node with and without TDC (Source: CiA 601-1)

Related articles

- ◆ [Automated validation of CAN FD networks](#)
- ◆ [CAN FD and the CRC issue](#)
- ◆ [CAN FD: Improved residual error-rate](#)
- ◆ [CAN FD simulation in real-time systems](#)
- ◆ [Secure communication for CAN FD](#)
- ◆ [Stand-by transceiver with fail-safe features](#)
- ◆ [Up to six CAN FD cores on one micro-controller](#)

parameter groups into CAN FD data frames. It provides an optional safety/security field. The CAN identifier contains the source address as specified in SAE J1939-21. It is intended to reuse the 8-byte parameter groups as specified in J1939-71. In the future, longer or shorter parameter groups could be specified and mapped into frames as specified in CiA 602-2.

The challenge: designing a physical CAN FD network

Specifying higher-layer protocols using the longer CAN FD frames is simple compared to the design of physical networks with transmission speeds higher than 1 Mbit/s. Do not underestimate the challenges in designing multi-drop networks running at higher bit-rates. Already 1 Mbit/s is a challenge. The automotive industry has not used Classical CAN networks with the maximum bit-rate in cars and trucks. Some truck makers use 667 Mbit/s or even 800 kbit/s.

If just two nodes communicate and all physical layer elements (I/O ports, connectors, cable, etc.) provide an impedance matching to the termination at both ends, there are no disturbances on the bus-lines.

Of course, you need dedicated connectors and cables. The limitation is just the speed of light, or more precise the maximum speed of electrical pulses in the cable. Theoretically, you can reach 10 Mbit/s at about 100 m.

If there are more nodes communicating, things become more complex. In theory, bus-line topologies terminated at both ends with very short not terminated stubs are the optimum. Star and hybrid topologies are more challenging, increasingly with higher bit-rates. From first experiences, star topologies with just a few branches are possible running with 2 Mbit/s. In these cases, the star center should be terminated with two 30-Ω resistors.

Topology is just one topic, when designing a physical CAN FD network. It starts all with the tolerance of the CAN controller's oscillator. The ISO 11898-1 standard provides five requirements, which you have to meet when calculating the allowed tolerance of the oscillator. To make it simple: Up to transmission speeds of 4 Mbit/s, the same tolerance as in Classical CAN guarantees a robust communication. For higher speeds more precise and more expensive oscillators are needed. In any case, it is specified in ISO 11898-1 that the frequency of the oscillator shall be 80 MHz, 40 MHz, or 20 MHz. When the oscillator is implemented by means of configurable cascaded PLL (phase locked loop) circuitries, the user should take care to configure them in a way, that the tolerance requirements are met. ▷

2015

30 years on from when Kvaser was founded, the world looks very different but communication is still our core mission.



Kvaser's engineers specialise in CAN, but we also value real human contact. Got a CAN problem? Talk to us and we'll find a solution.

To find out more:
www.kvaser.com/support
+46 31 706 1375



30 YEARS OF EXPERIENCE

Find CAN hardware
and software at
www.kvaser.com

YOUR PORTAL TO THE
PERFECT CAN SOLUTION

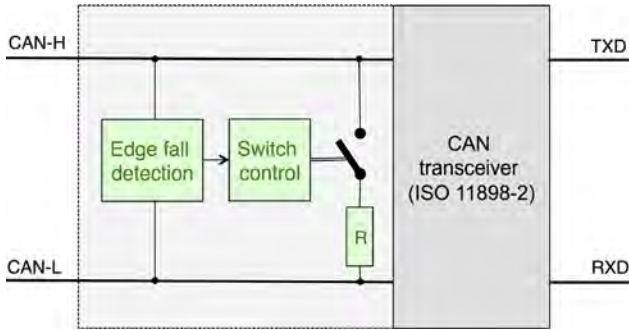


Figure 3: The ringing suppression circuitry specified in CiA 601-4 can be integrated into ISO 11898-2 compliant transceivers or can be used as stand-alone chip (Source: CiA 601-4)

Configuring the bit-timing is the next topic. Of course, the configuration of the arbitration bit-time is the very same as in Classical CAN. You have to set the bit-rate prescaler, the propagation segment (always one time quantum), the propagation and phase segment 1 (often just one setting), the propagation segment 2, and the synchronization jump-width. In order to minimize the quantization error, when switching to the higher bit-rate, the number of time quanta per bit should be large enough. Bosch's M-CAN module allows 385 time quanta per bit.

The detailed parameter setting for dedicated bit-rates is not in the scope of ISO 11898-1. CANopen FD (CiA 301 version 5.0) and CiA 602-1 (CAN FD physical layer for commercial vehicles) will provide such specifications for different bit-rates. Also SAE 2284-4 will specify the CAN FD bit-timing for passenger cars as well as some other details of the CAN FD physical layer design. The document is still under development.

The data-phase bit-timing, which is independent of the arbitration bit-timing, use the same parameters as mentioned above plus the transmitter delay compensations (TDC) and the TDC offset. The CiA 601-1 CAN FD physical layer design specification recommends the enabling of the TDC function.

The CiA 601-1 document provides some physical layer design rules for CAN FD nodes. The access is limited to CiA members. However, interested parties willing to review and comment the specification may request a personalized copy from CiA office. The confirmation of sample points for both bit-rates should be the very same in all nodes. This increases the robustness of the communication. It is also recommended to configure the bit-timings with a resolution meaning the maximum possible number of time-quanta, in order to reduce the quantization error. This leads to an increased phase margin. CiA also recommends using transceiver chips qualified for higher bit-rates. They feature less asymmetry, which remains more phase margin for the ringing caused by the chosen topology and other physical layer effects.

The transceivers feature different loop delays for the recessive-to-dominant and the dominant-to-recessive transitions. The symmetry of these two loop delays depends on the symmetry of the internal transceiver delays and on the resistive and capacitive busload. The asymmetry of the transceiver loop delay shortens or expands the bit in the arbitration phase and the data phase. Towards higher bit-

rates the symmetry becomes more and more important. The ISO 11898-2 specifies min/max symmetry values for 2 Mbit/s (-65 ns/+40 ns) and for 5 Mbit/s (-45 ns/+15 ns).

The symmetry of the RxD signal on the receiving node is defined by the symmetry performance of the transmitting and receiving node. To guarantee a robust communication between two or more nodes in a network, the transceiver Tx delay symmetry of the transmitting node and the transceiver Rx delay symmetry of the receiving node shall be very accurate. The loop delay symmetry cannot cover this in total. It is possible that one device in the network has a transceiver with a very symmetric transmitter part and an asymmetric receiver part. The behavior of a transceiver from another supplier may behave vice versa. Therefore, in the CiA 601-1 specification additional parameters were defined for the transmitter and receiver symmetry (e.g. $t_{REC}(RxD)_{min} = 110 \text{ ns}$ and $t_{REC}(RxD)_{max} = 225 \text{ ns}$ for 5 Mbit/s). The document also describes how to calculate the jitter bit length seen by the receiving node. These values consider only the influence of the transceiver. Additional effects such as clock tolerance and the phase shift of the network are discussed in /CiA601-3/, which is still under development.

The phase margin depends also on the oscillator frequency and some non bit-rate depend reasons. This includes the bit asymmetry caused by the transceiver and other physical layer elements und the unstable RxD signal caused by the ringing. The ringing on the bus lines comes from the used bus topology.

Network designers should also consider temperature-depend effects. For example, the isolation material of cables can change the impedance depending on the temperature. In the CiA 602-1 CAN FD physical layer specification for commercial vehicles, it is recommended to not PVC cables. Asymmetric PCB (printed circuit board) layouts or connectors can also cause ringing. Therefore, the CiA 602-1 specification for CAN FD physical interface for commercial vehicles proposes that the untwisted length of the wire in the area should be as short as possible (in maximum 50 mm). There are some more recommendations regarding electrical connector parameters.

Even the pinning of angular connectors is an issue at higher bit-rates. The CAN_H and CAN_L pins should have the same length.

In order to suppress ringing in star and hybrid topologies, special circuitries could be used. Denso's has submitted its ringing suppression technology to CiA for

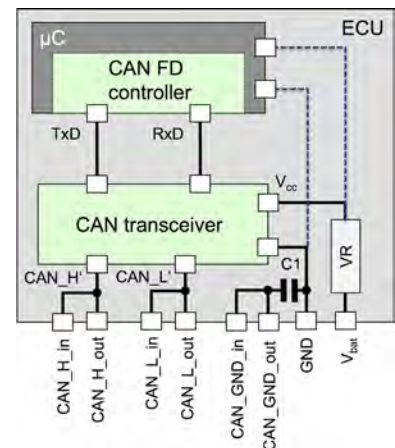
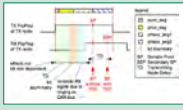


Figure 4: Typical interface of ECUs for commercial trucks as proposed in CiA 602-1 (C1 = 100 nF)

CAN Newsletter Online

The CAN Newsletter Online sister publication provides brief product-related information. For more details please visit www.can-newsletter.org.

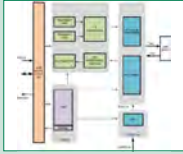


CAN FD

Recommended practice

CiA has released part 1 of the CiA 601-1 CAN FD node and system design

recommended practice. The series fills the gap between the ISO standards for CAN FD and the system design specifications. [Read on](#)

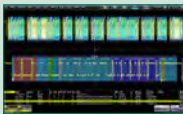


IP solution

Non-ISO CAN FD solution for automobiles

Arasan's Total IP Solution implements the Classical CAN protocol, as well as the non-ISO CAN FD protocol compliant

to Bosch. The company plans the development of CAN FD transceivers, too. [Read on](#)



Oscilloscope

CAN FD analysis on the symbolic layer

Teledyne LeCroy has announced the addition of symbolic (application) layer analysis capabilities to their CAN and CAN FD serial trigger, decode, measurement, and graphing solutions. [Read on](#)



CAN sniffer

Free tool with interface purchase

As a distributor of Kvaser interfaces in the UK, Warwick Control is introducing a new policy for its customers: Any Kvaser interface purchased from Warwick Control comes with a free ECO version of the X-Analyzer 3. [Read on](#)

standardization and submission to ISO. It is described in CiA 601-4, which is under development. The basic idea is to change the impedance of some nodes dynamically. It is switched on for a short part of the bit-time and then switched-off again. In CiA 601-4, the maximum start time is specified with 50 ns and the end-time between 200 ns and 410 ns.

General Motors prefers another solution: A bus-line topology with all nodes terminated locally. In March 2015, the intended GM wiring harness was tested successfully during the CAN FD plugfest in Detroit organized by CiA. GM likes to use 24-m bus-line networks with 20 nodes and not terminated stub-lines. The maximum stub-line is 1,7 m.

More experiences are necessary

CiA will organize next CAN FD plugfest, when new products are launched or additional network approaches are submitted for testing. Additionally, CiA will collect experiences and organize the exchange of knowledge within the IG CAN FD respectively the IG commercial vehicles. Additional participants are welcome. The upcoming CAN FD Tech Days will update newcomers with Classical CAN background in CAN FD technology. At the 15th international CAN Conference in Vienna in October (27 and 28) several papers are related to CAN FD topics.

Holger Zeltwanger

2015 and beyond

For the next 30 years, our mission is to continue developing innovative, high-quality interfaces for CAN and related buses.

1996 ● WAVEcan Bluetooth



1997 ● LAPcan PCMCIA



2005 ● Leaf Light USB



2012 ● Blackbird Wifi



2016 ● Air Bridge Radio

The industry's first point-to-point CAN connection over radio.

Replace the twisted pair with high quality wireless freedom.

More info: sales@kvaser.com

COMING SOON



30 YEARS OF INNOVATION

Find CAN hardware and software at

www.kvaser.com

YOUR PORTAL TO THE PERFECT CAN SOLUTION

Flexible and scalable CAN solutions

CAN controllers need to evolve to fit into Industry 4.0. With Xmos CAN controllers, designers can create a system-on-chip to meet their precise requirements, including multiple CAN controllers and Ethernet connectivity.

The CAN protocol has been around for over 20 years, and is still very popular in automotive and industrial applications. Several hundred million CAN nodes are sold each year mainly as single or dual-channel interfaces. CAN controllers are available as stand-alone CAN chips or are integrated into micro-controllers. But the recent trend in industrial applications towards Industry 4.0 means there is a need for CAN to be used in more complex systems. Conventional CAN controllers cannot meet the new demands that have come with advancements in technology. Existing CAN solutions are not scalable and do not allow system designers to expand the CAN channels or interface with other protocols like Ethernet, USB, or Industrial Ethernet. This article presents a flexible, scalable, and adaptable CAN solution that meets the requirements of Industry 4.0 systems.

The CAN protocol was designed originally for the automotive industry and is now used widely in industrial automation due to its robust nature, low cost, flexibility, and reliability. The rise of Industry 4.0, or the 'Connected Industry', will bring significant changes across the manufacturing and industrial sectors. The vision of Industry 4.0 is to realize the 'software-configured' factory. This requires connecting sensors, actuators, PLCs, and various embedded devices to the Internet in a safe way so that they can be monitored and managed in the Cloud. As the Industry 4.0 movement continues to grow, existing CAN systems need to be integrated into this new infrastructure. This presents many opportunities, but it also brings challenges in order to implement it. When the original CAN controllers were designed, people were not aware of these latest requirements and challenge: connecting CAN nodes to the Internet, scaling CAN-based systems across multiple sites, or managing distributed systems.

CAN controllers need to evolve to fit into Industry 4.0. One option is to design a custom chip, or to use an FPGA, but both are costly and not necessarily scalable. Xmos multicore micro-controllers provide a different solution, allowing designers to build their systems using a single off-the-shelf device that is easy to configure, flexible, and scaleable. Developers can use Xmos to build systems that fit precise configuration requirements, including single PLCs connected to multiple CAN nodes, single CAN nodes connected directly to Ethernet, or a series of CAN nodes concatenated together and linked to Ethernet.

The Xmos CAN controller is a software-based solution implemented using xCore multicore micro-controllers. The implementation exploits key features of the architecture of these multicore micro-controllers such as configurable I/Os, determinism and concurrent processing, as well as high performance scalability.

xCore architecture

xCore is a new class of micro-controllers that has multiple 32-bit processor cores, flexible I/Os, and a timing deterministic architecture that makes it very easy to use. Unlike conventional micro-controllers, xCore devices can run multiple real-time tasks simultaneously, allowing engineers to create complicated CAN-based systems.

The micro-controller is made up of multiple 'logical processor cores' distributed across one or more xCore tiles. The software running on the micro-controllers is completely timing deterministic.

Determinism is rooted in some of the fundamental architecture features like single-cycle instruction execution, interrupt-less executions, efficient I/O logic, and an RTOS-like hardware scheduler. The architecture has an integrated set of I/O ports controlled directly from the logical cores. The low latency ports are well suited for the realization of real time complex protocols such as CAN. They allow easy buffering of values on ports, which is critical in implementing CAN on the xCore architecture. ▶

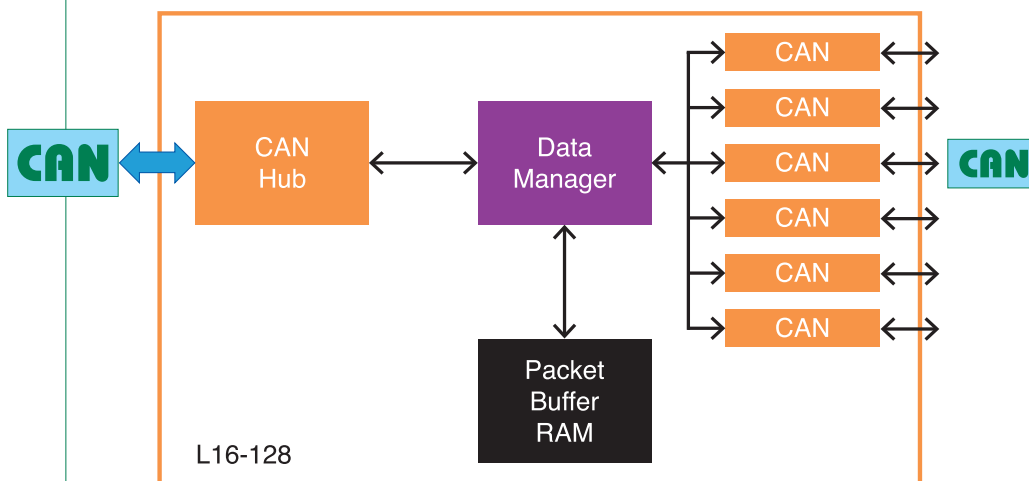


Figure 1: Multi-port CAN hub from Xmos (Photo: Xmos)



CAN in Automation

15th international CAN Conference

Vienna (AT), October 27 and 28, 2015

Profit from the presentations of CAN and CAN FD experts.

Conference sessions are on:

- ▶ Vehicle application
- ▶ Physical layer
- ▶ Migration to CAN FD
- ▶ Concepts of CAN
- ▶ Higher-layer protocols
- ▶ Security
- ▶ CAN FD design
- ▶ CAN and IoT

Benefit from networking and exchanging experience with CAN experts from different business areas.

Tabletop exhibitors are:

- ▶ Cypress Semiconductor
- ▶ esd
- ▶ Fraunhofer IPMS
- ▶ HMS
- ▶ Janz Tec
- ▶ Provertha
- ▶ Renesas
- ▶ Rohde & Schwarz
- ▶ Telemotive
- ▶ Vector

Register before September 25, 2015 to make sure you get the early bird rate.

*For more details please contact the CiA office
at marketing@can-cia.org*

www.can-cia.org

CAN on xCore

The software-defined CAN controller uses two 1-bit hardware-response ports that are configurable at compile time. A single logical core is all that is required to implement one CAN interface. The CAN controller is ISO 16845 compliant and is delivered as a software library that engineers can program onto an Xmos multicore micro-controller.



Users can create their own CAN system on a single chip by specifying the number of required CAN interfaces and also the bit-rate of each interface. Users can create up to 12 CAN interfaces on a single XS1 device. The flexibility in the software implementation allows users also to create CAN protocol bridging with Ethernet, USB, and other protocols.

In addition, designers can create a heterogeneous system on the different processor tiles. For example, designers can create a CAN hub on one tile, which collects data from multiple CAN interfaces and a CAN-to-Ethernet gateway on the other tile. The CAN hub collects the data from the different CAN interfaces running at the same or different bit-rates and passes it through an integrated hardware switch to the CAN-to-Ethernet gateway application on the other tile, which then passes the collected data through Ethernet to a remote node. Using xCore makes it possible to create complex systems on a single chip.

The xTimecomposer development tools allow for customization of the CAN library to add additional application specific features. While a CAN node runs in one logical core, other applications can run in other cores and communicate with the CAN node using the API provided with the CAN library. In addition to the CAN hub and protocol converters, the CAN controller on the multicore micro-controllers can be used to act as a CAN sniffer. The CAN sniffer only listens on the CAN network without acknowledging the CAN packets.

Almost all existing CAN implementations use interrupts to notify applications of incoming and outgoing CAN frames. CAN applications have to handle these interrupts in real-time along with other tasks, which makes it challenging to handle worst-case situations particularly in high complicated systems. The software-based Xmos CAN solution uses events and Fifos where the received messages are stored for application tasks to consume instead of interrupts; hence it is deterministic by design.

The CAN solution supports bit-rates from 31,25 kbit/s to 1 Mbit/s. For a bit-rate of 1 Mbit/s, 100 MIPS (million instructions per second) is required; for bit-rates of 500 kbit/s or lower, 62,5 MIPS is required. The XS1 family of devices that run the CAN solution are available with up to 1000 MIPS compute, providing a range of single and dual tile devices that can be used for CAN systems.

Application examples

Multi-port CAN hub: The dual-tile XS1-L16-128 micro-controller allows you to program the exact number of CAN interfaces and the bit-rate you need for your design. It is easy to create a multi-port CAN hub on a single multicore micro-controller. The XS1-L16-128 micro-controller provides up to 8 CAN ports in total at 1 Mbit/s (4 per tile). In case of lower bit-rates, 12 CAN ports are provided in total (6 per tile). The ad-

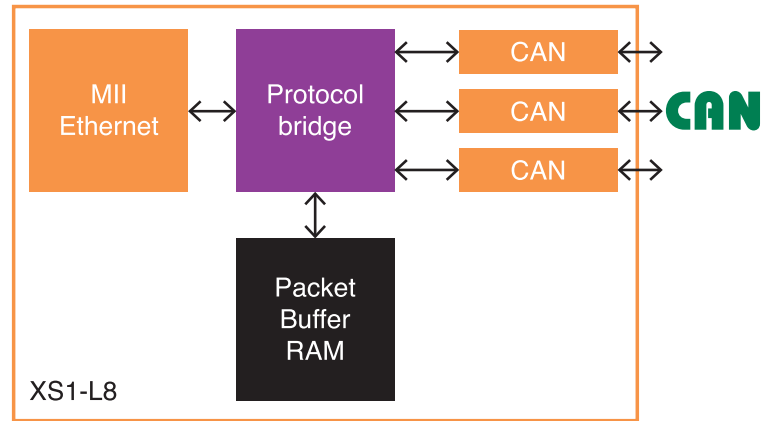


Figure 2: CAN-to-Ethernet hub from Xmos (Photo: Xmos)

ditional logical cores are available for data manager and application software.

Ethernet to CAN bridge: Xmos micro-controllers can be used for bridging different protocols. For example: transmitting data from multiple CAN devices over Ethernet. The dual-tile XS1-L8-128 micro-controller provides up to 3 CAN ports at 1 Mbit/s, Ethernet MAC, and an MII interface. Two logical cores are available for protocol bridging and customer applications. More channels can be added if slower CAN interfaces are required. This Ethernet to CAN bridge is suitable for Industry 4.0 CAN-based solutions. It is well suited for manufacturing environments where data from multiple CAN interfaces has to be accessed or controlled from a remote platform through Ethernet. The Ethernet interface could equally be an Industrial Ethernet standard such as Profinet RT or Ethernet/IP.

Summary

With the advancements in Industry 4.0, embedded devices must provide more functions to offer maximum customer benefit. However, most CAN solutions are not sufficiently sophisticated when users try to expand their systems to multi-channel solutions or build a CAN-based solution for Industry 4.0. With the conventional CAN controllers it is not possible to build a single complex CAN-based system on a single chip. Also, such solutions cannot be scaled easily to larger systems or combined with other industrial protocols.

With Xmos CAN controllers, designers can create a system-on-chip to meet their precise requirements, including multiple CAN controllers and Ethernet connectivity. They can choose the number of CAN interfaces and the bit-rate of each interface. The CAN solutions bridge the gap between challenging requirements from Industry 4.0 and the limitations in the conventional CAN controllers.



Author

Shanthini Kannan
 Vinith Kumar Mundhra
 Xmos Semiconductor India Pvt Ltd.
www.xmos.com



CAN in Automation

The non-profit CiA organization promotes CAN and CAN FD, develops CAN FD recommendations and CANopen specifications, and supports other CAN-based higher-layer protocols.

Join the community!

- ▶ Initiate and influence CiA specifications
- ▶ Receive information on new CAN technology and market trends
- ▶ Have access to all CiA technical documents also in work draft status
- ▶ Participate in joint marketing activities
- ▶ Exchange knowledge and experience with other CiA members
- ▶ Get the CANopen vendor-ID free-of-charge
- ▶ Get credits on CANopen product certifications
- ▶ Get credits on CiA training and education events
- ▶ Benefit from social networking with other CiA members
- ▶ Get credits on advertisements in CiA publications

*For more details please contact the CiA office
at headquarters@can-cia.org*

www.can-cia.org

ODX-based flash solution

The described flash solution was approved in practice using an ECU for construction machines. The powertrain ECUs exchange data via a CAN-based J1939 network.



Figure 1: Liebherr Mining Excavator R995 without license to operate on open public highways

Non-road mobile machines (NRMM) that are powered by heavy-duty diesel engines with emission control systems are controlled by embedded electronic control units (powertrain ECUs). These powertrain ECUs exchange data via a CAN-based J1939 network and are connected to control units that are especially made for NRMM applications. Modern control units are connected to the powertrain CAN and support a diagnostic protocol that comes with the capability of reprogramming the flash memory.

A control system consists of one or more embedded ECUs that process data according to the IPO (input-processing-output) principle. On NRMM, typical input data are physical values that are measured by sensors (e.g. temperature, pressure, revolution speed). Typical output data are for drive displays, lamps, or electro-hydraulic components, such as on/off or proportional solenoid valves. The NRMM control system consists of one or more embedded ECUs. The basic architecture of an embedded ECU consists of a central processing unit (CPU), random access memory (RAM), read-only memory (ROM), inputs, outputs, power supply, and a connection to the in-vehicle network – in this example CAN.

NRMM ECUs usually employ a CAN-based communication technology specified in the SAE J1939 set of recommended practices. The ECUs of the engine, the transmission, and the emission control system are part of the J1939 network. For maintenance and service, CAN provides access to the powertrain and the entire control system even if the ECUs are installed somewhere in the machine. External test equipment (a tester) is connected to the CAN network of the machine (Figure 3). If both the tester and the ECU support the same diagnostic protocol, the tester can send a diagnostic service request to an ECU and the ECU answers with either a positive or a negative response. This kind of data communication, usually referred to as diagnostic communication, is specified in a diagnostic protocol, such as UDSonCAN. UDSonCAN is short for Unified Diagnostic Services on Controller Area Network and is specified in ISO14229.

External test equipment components

Figure 3 illustrates how external test equipment is connected to the machine: A vehicle communication interface ▷

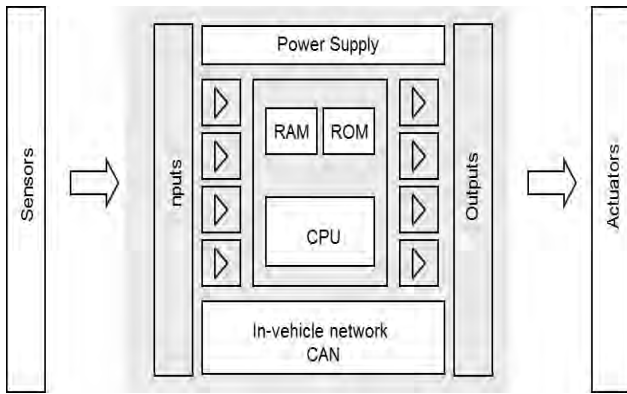


Figure 2: Basic architecture of an embedded ECU

(VCI) is connected to the USB port of a PC, and the VCI is connected to the DLC (data link connector), which provides access to the CAN network. Figure 4 illustrates the software components of a tester. In the context of this article, the application, the D-Server API, the MVCI D-Server, the D-PDU API, and the ODX data are of importance.

ODX is short for Open Diagnostic Data Exchange and specified in ISO 22901. The ODX data file contains the description of the ECUs in an internationally standardized format. It contains at least the diagnostic protocol(s) and the communication parameters for the diagnostic communication, e.g. for UDSonCAN. The MVCI D-Server executes the application and processes the ODX data. It also connects the application with the VCI, and therefore with the ECU(s).

The application talks to the ECUs by sending diagnostic service requests and receiving the responses. Depending on the use case, different tester applications can be created on the same set of D-Server and ODX data.

Today, the capability to reprogram ECUs is a must in all phases of the machines' life cycle. For this purpose, the MVCI D-Server contains a job processor for the execution of Java jobs and a flash data processor that is specialized to support the programming of flash memory. Alternatively to Java, the reprogramming application can be created as OTX sequences or simply as C/C++ code. In any case, the result is an ODX-based flash solution (flashtool) for NRMM control systems.

Implementation example

The Compact Control Unit CCU 70 of Liebherr Elektronik is a programmable ECU, which was specially developed for applications in mobile machinery and commercial vehicles. These use cases require reliability even under extreme environmental conditions such as vibration, dirt, humidity, salt spray, or electromagnetic influence. The IP rating of the connected ECU is IP6K9K. The CCU is available with three CAN and one Ethernet interfaces, up to 70 inputs and outputs, and it comes with 4-MiB flash memory. CCU 70 supports both J1939 for the on-board communication with other ECUs (e.g. another CCU or the engine controller) and UDSonCAN for diagnostic communication with external test equipment.

LIN & CAN Bus simulation for test and production

LIN protocol >= 2.0 only:

Produkt-ID des Herstellers:

Funktion:

Variante:

Antwortfehler:

P2 min [ms]:

ST min [ms]:

N_As Timeout [ms]:

N_Cr Timeout [ms]:

I/O Digital

RS-232

Distribution China: Hongke Technology Co., Ltd Ph:86-400-999-3848 sales@hkaco.com www.hkaco.com

Distribution USA: DGE Inc. Ph: 248-293-1300 sales@dgeinc.com www.dgeinc.com

Lipowsky Industrie-Elektronik GmbH Ph:+49 6151-935910 info@lipowsky.de www.lipowsky.de

**LIPOWSKY
INDUSTRIE-ELEKTRONIK**

ODX data and flashware

Figure 6 illustrates the 8 categories of an ODX database. The ODX category Flash is the only ODX database category that is solely used for reprogramming. The other ODX categories are also used for other diagnostic applications. The most important categories are the Communication Parameter Specification and the Diagnostic Layer Container (DLC). Figure 6 shows the internal structure of a Diagnostic Layer Container. Because ODX describes the data format, not the content, it depends on the author if and how the ODX categories are used. For the description of a single ECU, at least a Communication Parameter Specification, one DIAG-Layer, and – for reprogramming – the ODX category Flash must contain data for the D-Server. For the purpose of processing by the D-Server, the ODX data has to be converted in a binary runtime equivalent (*.sod). The runtime database with the description of the CCU70 is named CCU70.sod.

ODX data files can contain one or more ECU descriptions, even with different diagnostic protocols (communication parameters, request/response definitions). For the purpose of diagnostic communication, the MVCI D-Server not only needs to know the parameters of the VCI and the physical location of the ECU in the vehicle, but also the location of the ECU in the database. This information is provided as a Logical Link in the application software of the flashtool.

Flashware is the data that is programmed in the flash memory in the end. It contains executable machine code and configuration data. The ECU application software programmer creates a source code and uses a compiler or assembler to convert the source code to machine code, which is a HEX file that is then used to be programmed to the flash memory by a flashtool. The data format of the flashware can be binary, Intel Hex, or Motorola S.

The ODX category Flash for the CCU contains the definition of flashware with the data format Motorola S, usually referred to as S-records. S-record files can be identified by their file extension name *.s19. Figure 7 shows a small excerpt of the CCU S-record file. In this example, the data field contains FFh, meaning that the records do not contain executable machine code. The first two characters of the S-record (S3) contain the information that this record has a 4-byte memory address and contains data. The two characters after the S3 are the hex-coded byte-count. The byte-count contains the number of bytes that follow the byte-count. Here, the byte-count reads 25h (37d), which stands for the 4-byte memory address, 32 data bytes, and 1 byte for the checksum.

Only the 32 data bytes per S-record are programmed to the flash memory. The memory address ends with 00h, 3Fh, FFh, FFh, representing the 4 MiB of flash memory. The starting memory address is 00h, 10h, 00h, 00h, meaning that the reprogramming covers only 3 MiB of the available 4 MiB flash memory.

Reprogramming sequence

If there is only one CCU connected to the CAN network, preconfigured CAN-Identifiers for the request and response services can be used by the flashtool and the CCU. In this

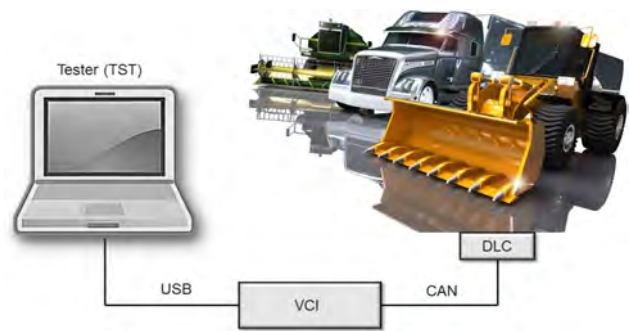


Figure 3: External Test Equipment is connected to the CAN of the machine

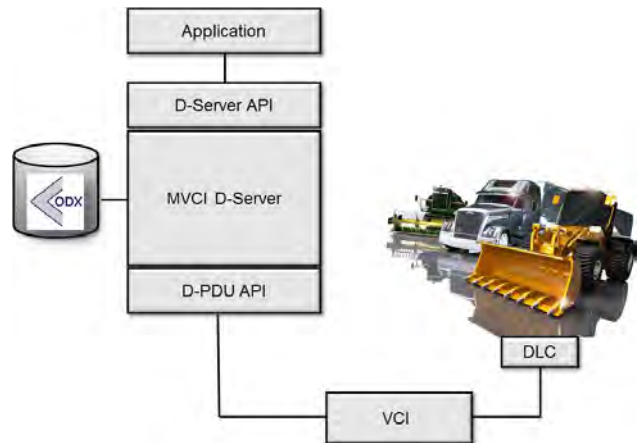


Figure 4: The software components of external test equipment

example, 29-bit CAN-Identifiers are used. In the following, the reprogramming sequence for a control system with one CCU is described: The first step is to start the LICoS Flashtool. The flashtool asks the user to switch the CCU's power supply off and on. The CCU boots and the boot loader starts reading the CAN messages. It looks for a so-called Force Download (FD) message (FD1 or FD2) that is sent on the CAN network by the flashtool.


FD1 is used if the control system contains only one CCU, FD2 if the control system contains more than one CCU. The FD message has to be sent by the flashtool with a cycle time of at least 10 ms. If the CCU receives either FD1 or FD2 within 50 ms, the boot loader firmware prepares the reprogramming sequence. If the boot loader receives the FD1 message, it sets an output of the CCU (LED blinks) and waits for the first UDS request, which is the request to transition into the extended diagnostic session.

The CCU answers the request with a positive response and transitions into the extended diagnostic session. In that session, the CCU requires a security access. For that purpose, the flashtool sends a request to get the seed (securityAccess > requestSeed). The CCU answers with a positive response and sends the seed as a 64-bit data parameter in the response. The flashtool takes the seed and calculates the key using a C-DLL that contains the seed-key-calculation algorithm.



Figure 5: Liebherr Compact Control Unit CCU 70

International trade magazine
for the technology of elevators
and escalators

 lift
report

PLEASE
CONTACT
US TO GET
A SPECIMEN
COPY!

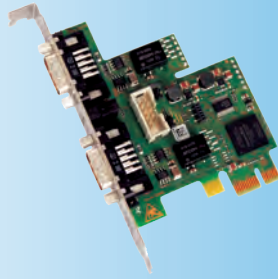
**Lift-Report takes part regularly in all major
fairs and trade events – worldwide.**

**Take the advantage of the magazine and make
new contacts in the lift industry.**



Hengsener Straße 14 . 44309 Dortmund . Germany
Phone: +49(0)231/92505550 . lift@vfz-verlag.de
www.lift-report.de

CAN-PCIe/402



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



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



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

Ethernet



CAN-PCI104/200

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

EtherCAN/2

- 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-CANopen
- PROFIBUS-CANopen
- PROFIBUS-DeviceNet
- EtherNet/IP-CAN

CAN-USB/2



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

Gateways



Visit us at the
sps ipc drives
 Nürnberg, 24. - 26.11.2015
 Hall 2, stand 130





Solutions for ▶▶ CAN

First class solutions for your CAN and CAN FD based projects

Apply the complete Vector tool chain to increase the efficiency of your projects:

- > Tools for testing, flashing and calibrating ECUs
- > Flexible bus network interfaces
- > High performance Scope for bit accurate signal analysis
- > Easy to configure AUTOSAR basic software
- > Worldwide engineering services and trainings

More CAN power: benefit from Vector's 25 years of CAN experience

▶ Information and downloads: www.can-solutions.com

▶ **CAN/CAN FD Poster**
now order for free:
www.vector.com/canfd_poster