December 2015

CAN Newsletter

Hardware + Software + Tools + Engineering

15th iCC: Focus on CAN FD

Plug-and-secure communication for CAN

Customized CANopen tools made easy

EMC effects underestimated as fault causes

Engineering

www.can-newsletter.org

PCAN-Router FD
Programmable CAN FD Router
NEW
PEAK System

Applications

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

Engineering

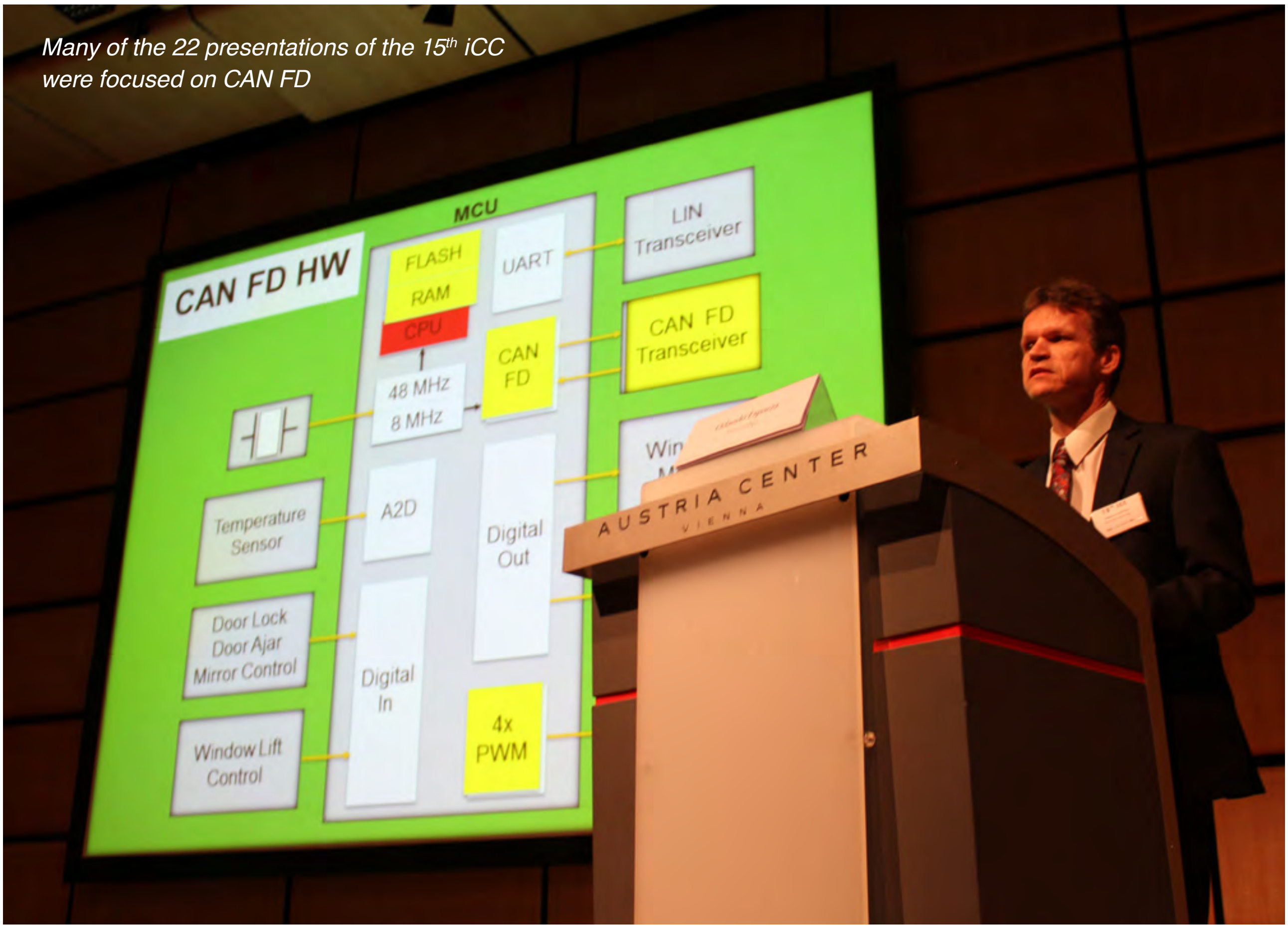
Software

CAN Newsletter Online

While working on this edition of the CAN Newsletter, we are always also busy writing for its sister publication the CAN Newsletter Online. If you are looking for more regular updates from the world of CAN then this website is the place to go. If you got any CAN-related product, we are always happy to receive and publish your press releases, datasheets, articles, and so on. Just send us an email.

15th iCC: Focus on CAN FD

About 100 participants listened to 22 papers. No doubt, the CAN FD related presentations were of most interest. Other hot topics were security and the Internet of Things (IoT).



Many of the 22 presentations of the 15th iCC were focused on CAN FD

The 15th iCC conference took place in Vienna (Austria). The first session introduced into some unusual vehicle applications. Oliver Hrazdera from Rosenbauer reported about the usage of CAN networks in fire-fighting trucks. Several CAN networks are implemented, which are partly interconnected by means of bridges and gateways. For future functional extension (e.g. secure communication), he requested more bandwidth. Gennady Benderman from Porsche came to the same requirement. He presented future electric/electronic (E/E) architectures for front-lights. For future high-resolution headlamps with thousands of pixels using a deeply embedded CAN network, the bandwidth of Classical CAN comes to its limits, because Porsche plans to control the LEDs pixel-wise. James Meer from Microflight reported briefly in an ad hoc presentation about CAN FD standardization in aviation. The responsible working groups in USA decided recently to support CAN FD in the Arinc 825-4 specification.

CAN FD related sessions

Magnus-Maria Hell from Infineon provided an update and summary on the discussion made in the last one-and-a-half year. He continued were he stopped at the last iCC. In particular, he explained the parameters introduced in the new ISO 11898-2 high-speed transceiver standard. This standard merges ISO 11898-2, -5, and -6. He explained the consequences for the partial networking respectively the selected wake-up option. Additionally, he discussed some network design options to reduce the ringing, e.g. to terminate all nodes. Another possibility to decrease the ringing in CAN FD networks was presented by Denso. Yuuki Horii introduced a ringing suppression circuitry (RSC) based on the idea to change dynamically the overall impedance of the CAN FD network. This RSC is specified in CiA 601-4, which will be released soon.

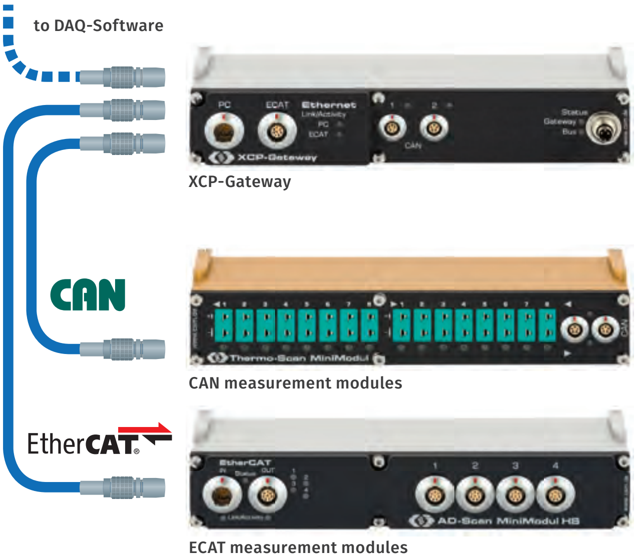
▷

Of course, you can migrate from Classical CAN to CAN FD strict forward by substituting the CAN hardware, controllers, and transceivers, in all nodes to make benefit of the higher throughput and longer frames. But some users require a more soft transition. NXP introduced in Vienna its FD Shield transceiver, which hides the CAN FD message to the Legacy CAN controller. Tony Adamson promoted the FD Shield as an interim solution for a quick integration of Legacy ECUs and CAN FD nodes in one network as well as a long-term solution for markets, in which a separation of CAN FD and Classical CAN communication is not desired due to the additional costs for the bridge/router device. A similar approach was presented by Kent Lennartsson from Kvaser: His CAN-FD Filter transceiver transforms the CAN FD message to a Classical CAN message with no data content. These two transceiver solutions seemed to be more suitable than the partial networking approach. This is to “switch-off” the Legacy CAN nodes during the CAN FD communication and to wake-up them again. The automotive industry is not in favor of such migration options. The carmakers prefer to migrate completely to CAN FD, in network segments, where higher bandwidth is required and to link those segments by bridge/router devices to segment running Classical CAN communication.

Even if you would like to substitute the entire CAN hardware by CAN FD capable semiconductors, you may need a migration path, because not all micro-controllers support currently CAN FD. Therefore, Wilhelm Leichtfried from Microchip presented a stand-alone CAN FD controller. He said that such external chips can help to minimize development timelines and can be more cost-effective than using high-end MCUs with CAN FD support.

The developers of software for CAN FD have already started to migrate to CAN FD – in particular, to frames with up to 64-byte payload. Dr. Oliver Hartkopp from Volkswagen presented a comprehensive survey about the integration, configuration, and usability of CAN FD in the Linux operating system. He gave an insight how programming interfaces have been altered in Linux in an evolutionary way without putting the existing application programming concept into question. Some of the presented ideas may be reused in other embedded setups – some may be too Linux specific to do so. He also presented the open source implementation of the ISO Transport Protocol as standardized in ISO 15765-2 supporting CAN FD frames. Peter Decker from Vector introduced the dynamic Multi-PDU-to-frame mapping. This approach requires in some cases more bandwidth than an optimized mapping. In the other hand, it is easier to test complex systems and more important it allows very flexible system designs and is supported by Autosar. Holger Zeltwanger presented the CiA 602 application layer, which maps the current SAE J1939 application layer to CAN FD frames. The proposed mapping complies with the Multi-PDU-to-frame mapping introduced by Vector. The CiA 602 proposal is already approved by simulation and requires without optimization just a third of the bandwidth when using Classical CAN.

▷



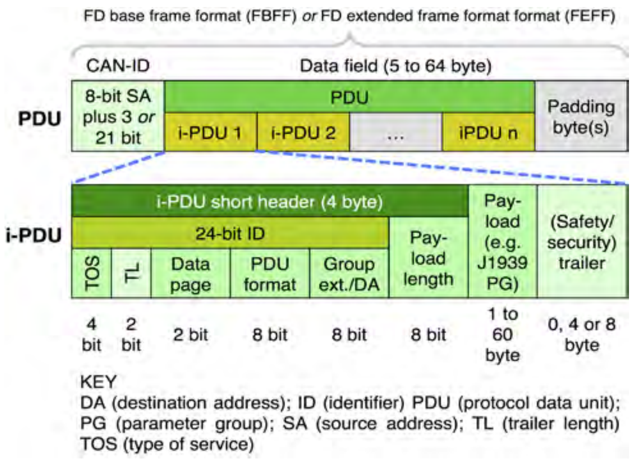

FD base frame format (FBFF) or FD extended frame format format (FEFF)
CAN-ID
Data field (5 to 64 byte)
PDU
8-bit SA plus 3 or 21 bit
PDU
i-PDU 1
i-PDU 2
...
iPDU n
Padding byte(s)
i-PDU
i-PDU short header (4 byte)
24-bit ID
TOS
TL
Data page
PDU format
Group ext./DA
Pay-load length
Pay-load (e.g. J1939 PG)
(Safety/ security) trailer
4 bit
2 bit
2 bit
8 bit
8 bit
8 bit
1 to 60 byte
0, 4 or 8 byte
KEY
DA (destination address); ID (identifier) PDU (protocol data unit); PG (parameter group); SA (source address); TL (trailer length) TOS (type of service)

Figure 1: CiA 602-2 frame mapping multiple SAE parameter groups into a CAN FD message (Source: CiA)

System design is the most challenging topic in CAN FD. Dr. Marc Schreiner from Daimler provided some general rules and recommendations for the physical network design. He discussed the influencing parameters and critical values of typical components. The main part of his presentation was focused on topologies and the possible bit-rates. He reported about his measurement in different CAN FD topology structures. In his work, the CAN FD signal asymmetries have been analyzed based on the RX as well as on a virtual RX signal based on the differential bus signal. Although the number of assessed variations was huge (approximately 750 in total) of course they cannot cover all kinds of CAN FD topologies that might occur in the field. Nevertheless the presented measurement results gave a good basic overview about the typical behavior of particular topologies and they might be a good help for a CAN FD system designer to configure networks in an appropriate manner. It should be noted that the presented results were valid only at room temperature. Significant changes over temperature have to be expected if PVC cable is used. He stated that the other main impact comes from the transceivers, which can be accounted for by applying the worst-case ISO values. If the CAN FD system design is targeting at 2 Mbit/s and higher in the data phase, he expected that the best results will be achieved with the point-to-point and with the line topology. Especially for conservative system designers, who do not want to tolerate the uncertainty of ringing in the network, the pure line topology is the only safe choice according to the Daimler researcher. Of course, short stubs can be used – but as short as possible, Dr. Schreiner said. According to him, ferrite star topology may work as well, when the branch length is short. The presented graphs are a good source of orientation.

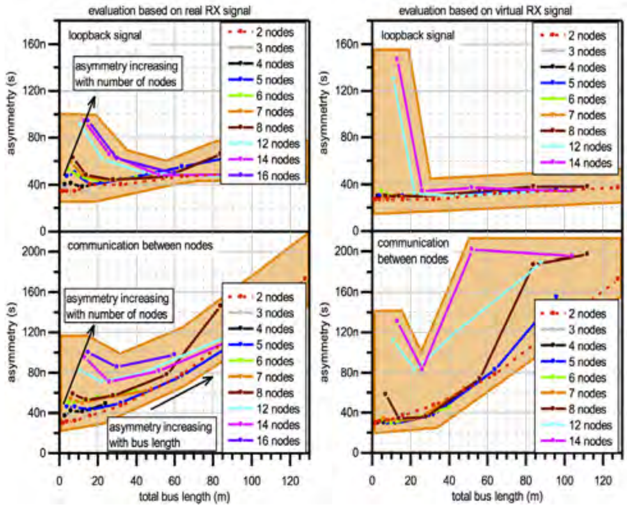


Figure 2: Asymmetry in a CAN FD network with line topology and equally spaced nodes (Source: Dr. Marc Schreiner, Daimler)

The University of Brighton presented the simulation results using the SAE benchmark message set. The performance analysis was realized for worst-case message delays, average message delays, and bus utilizations. The worst-case message delay analysis has shown that with the CAN FD model based on the SAE benchmark message set, from 1,78 to 3,28 times smaller worst-case message delays on average can be achieved compared to the Classical CAN model. Similarly, from 1,67 to 4,16 times smaller mean message delays on average has been observed with CAN FD. In bus utilization results, almost half the bus utilization values have been observed with CAN FD. The simulation results have revealed that the CAN FD protocol provides considerable performance improvements in message transmission speed and bus utilization compared to Classical CAN.

In order to improve the throughput, Gianluca Cena from the CNR-IEIIT institute presented a method to prevent bit stuffing. The proposal is suitable for CAN FD as well as Classical CAN communication. The zero stuff-bit (ZS) mechanism uses a codec, which can be implemented in software or in hardware. One of the benefits is that there is no time jitter of the messages depending on the content. Each message type has always the same length.

Dr. Arthur Mutter from Bosch explained in detail the performance of the improved error detection mechanisms in the CAN FD protocol. The four main improvements in CAN FD regarding error detection mechanisms are: use of CRC polynomials of higher order, inclusion of dynamic stuff bits into CRC calculation, inclusion of the number of dynamic stuff bits into the frame, and the use of fixed stuff bits in the CRC field of the frame. The results are mainly in the interest of attorneys. They are a proof that CAN FD is more reliable than Classical CAN.

Security and IoT

The two other important topics on the iCC were the Internet of Things (IoT) and secure communication. Amir Spivak from Beyond Security opened the discussion with a proposal to test third-party products and applications for unknown vulnerabilities. He used a black box test environment. This means he does not need the source-code of the ECU under test. In his research he found out that proprietary protocols are the weakest approaches to withstand attacks: "The writers of these protocols believe that no one can attack the protocol since its format is unknown. They are very wrong: it's possible to ▷

CAN Newsletter Online

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

15th iCC

Focus on CAN FD

About 100 attendees listened in Vienna to 22 presentations and one ad hoc paper. Many talks were related to CAN FD technology.

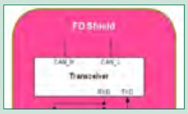
Read on

CAN FD products

Microchip goes automotive

Microchip has pre-announced its CAN FD stand-alone controller and transceiver. Its daughter company, K2L has launched interface devices supporting CAN FD to be used with the company's software tools.

Read on

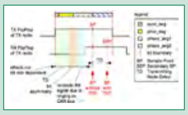


Transceiver

Filters CAN FD messages

NXP has presented an idea on how to run Classic CAN controllers in a CAN FD network. It is based on a transceiver able to filter CAN FD messages and also the Error frames caused by the Classic CAN chip.

Read on



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 Solultion implements the Classical CAN protocol, as well as the non-ISO CAN FD protocol compliant to Bosch. The company plans the developement of CAN FD transceivers, too.

Read on

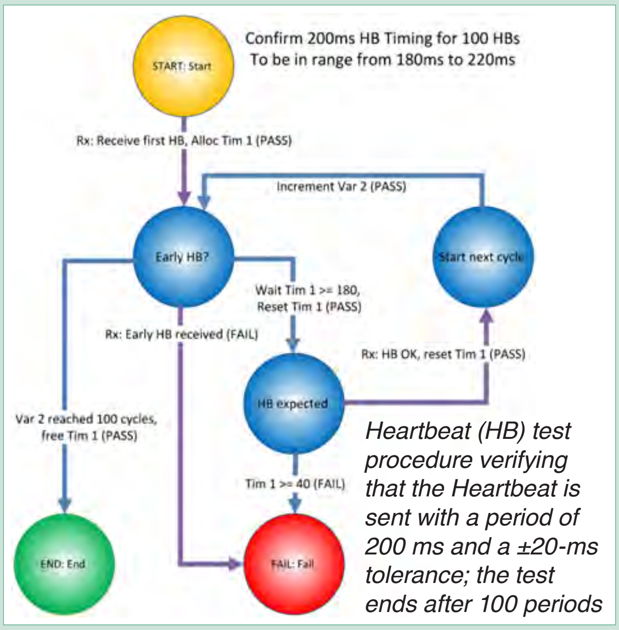
attack (as demonstrated by the black box test approach) and when attacked, weaknesses are immediately exposed."

Dr. Andreas Müller from Bosch presented for the first time an approach for establishing and refreshing symmetric cryptographic keys between different nodes in a CAN network in a plug-and-play manner. "Our approach captivates by its simplicity, low complexity and high cost-efficiency, and may be readily implemented without any modifications of off-the-shelf CAN controllers," he stated (see also the detailed article on page 10). Thilo Schumann from CiA presented the first results of a secure communication approach in CANopen networks. The proposed solution is based on authentication instead of encryption, because the nodes do not sent secret data, but should not accept commands from "strangers".

Secure communication is needed, when a wireless communication links two CAN segments. Derek Sum from Kvaser presented a wireless replacement for

Testing of CiA 447 devices

Olaf Pfeiffer from ESA presented at the iCC 2015 the test of CANopen networks used in special-purpose cars. The related CiA 447 application layer, recently submitted to ISO for international standardization, is suitable for taxis, police cars, ambulances, and car for handicapped drivers. The introduced CANopen Test Machine was jointly developed with Daimler. It is based on Microsoft Vision. The presented test tools can be used for individual test plans and the CiA 312 test plan, which is under development. In general, tests implemented based on the CANopen Test Machine operate directly on the DUT (device-under-test) with no other devices connected. However, in order to test a CiA 447 gateway, an additional tester device as specified in CiA 447 must be present. The test machine test can then send commands to the tester device to produce directed or non-directed background traffic of a specified load.


START: Start
Confirm 200ms HB Timing for 100 HBs
To be in range from 180ms to 220ms
Rx: Receive first HB, Alloc Tim 1 (PASS)
Increment Var 2 (PASS)
Early HB?
Start next cycle
Wait Tim 1 >= 180,
Reset Tim 1 (PASS)
Rx: Early HB received (FAIL)
Rx: HB OK, reset Tim 1 (PASS)
HB expected
Var 2 reached 100 cycles,
free Tim 1 (PASS)
Tim 1 >= 40 (FAIL)
END: End
FAIL: Fail

Heartbeat (HB) test procedure verifying that the Heartbeat is sent with a period of 200 ms and a ±20-ms tolerance; the test ends after 100 periods

cables in CAN network systems. The idea is not new. He discussed the general requirements on security and timing behavior, especially the real-time behavior, as well as the data consistency.

Secure communication is also required when cloud-based maintenance and services should be used. Dr. Heikki Saha from TK Engineering shared his thoughts on the example of CANopen-based control systems. He reported about first CANopen pilot projects proofing the proposed system integration framework for management process, device, tool, and service ecosystem into a web-based cloud environment. Further development in the near future will concentrate on the improvement of information logistics from design into assembly line and field service. Configuration packets are already automatically generated from corresponding CANopen design projects, but measurement setup creation from the projects shall be implemented. One of the missing links for diagnostics is the mapping of CANopen services to the ODX standard used by the automotive industry. This would be helpful in particular for the mobile machine industry and other off-highway vehicle fleets.

Erik Halvordsson from HMS Industrial Networks talked about the Industrial Internet of Things (IIoT). He gave examples (e.g. straddle carriers getting remotely orders via the web) on today's integration of CAN-controlled systems into the IT world. Another example was predictive machine analysis using CANopen data streamed to SAP's web-based database.

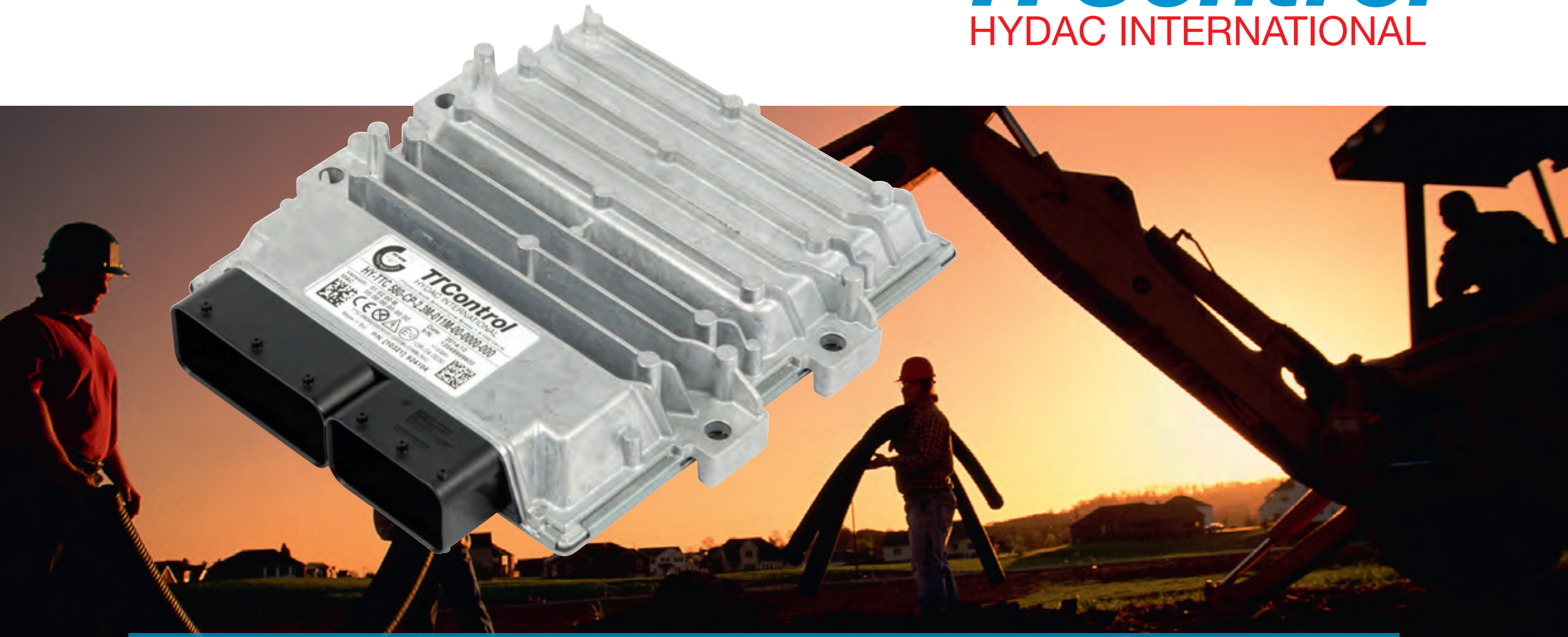
All papers are documented in the 15th iCC proceedings, which can be purchased from CiA office in Germany. The presentation slides are available for participants and buyers of the proceedings as PDF files.

The tabletop exhibition

As usual, the 15th iCC was accompanied by a tabletop exhibition. Besides already launched products, the participating companies presented also some new semiconductors, devices, and tools. Of course, everyone was looking for CAN FD controllers. Microchip presented its stand-alone CAN FD controller and its MCP2561(FD) transceiver family. The products will be available via distribution in the next year. Microchip plans also a single-chip comprising CAN FD controller plus transceiver. Renesas showed first samples of its RH850/F1K micro-controller featuring six CAN FD modules and an additional Classical CAN module. These chips suitable for gateway solutions will be available also in 2016. Cypress presented in Vienna its Traveo MCU family with CAN FD on chip. All these products support the ISO CAN FD protocol. The Fraunhofer IPMS exhibited its CAN FD core, which complies also with ISO 11898-1:2015. This core supports the non-ISO CAN FD protocol, too. Vector and K2L presented their tools supporting Classical CAN and CAN FD. ◀

Holger Zeltwanger

TTControl
HYDAC INTERNATIONAL



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

Plug-and-secure communication for CAN

Security is a topic of rapidly increasing importance in automotive as well as industrial applications. With Bosch's novel approach, symmetric cryptographic keys between different nodes in a CAN network can simply be established and refreshed.

The recent trend towards ubiquitously connected systems – be it cars, factories, or buildings – does not only come with numerous opportunities and benefits, but imposes also new security threats with a potentially huge impact. If everything is interconnected and APIs are introduced in order to facilitate innovative services and applications, the attack surface for malicious manipulations and intrusions is increased significantly. Without proper countermeasures, hackers may remotely control cars, eavesdrop on confidential production data, or manipulate a building automation system.

This threat is reflected by various prominent attacks that have been performed and published recently. In "Remote exploitation of an unaltered passenger vehicle" [1], for example, the authors describe how they managed to remotely inject messages to the CAN network of a car and thus affect important physical systems, such as steering or braking. The hackers connected to the vehicle via a mobile network. This led to a recall of about 1,4 million cars and fueled legislative initiatives to mandate car manufacturers to support reasonable measures to protect cars against hacking attacks [2]. More security leaks and attacks on cars and other vehicles have been reported reported [3]-[5]. Remote attacks may easily scale and hackers do not need physical access to the system under attack. For example, imagine a scenario with thousands of cars remotely hijacked. Hackers could precipitate a breakdown of the whole traffic infrastructure of a city or country by manipulating all cars in a coordinated manner. Clearly, this could not only lead to tremendous physical damage, but also to a significant impact on the whole economy and society. Therefore, the support of appropriate security mechanisms represents a crucial prerequisite for the success and acceptance of any connected system.

In automotive networks, a secured CAN communication represents a particularly important building block of security concepts, since a compromised CAN network can have a direct impact on people's safety. When CAN was introduced in the 1980s, security was not a crucial topic yet, since systems were closed and isolated. With the increased openness, however, this changes fundamentally. While in principle suitable concepts and algorithms are available [6], [7], they are not used yet because other challenges still remain. This includes proper standardization across different manufacturers and suppliers, but also efficient approaches for establishing, refreshing, and managing the cryptographic keys that are required for the involved cryptographic schemes.

We therefore propose a novel approach addressing the latter aspect: this approach is able to establish and refresh symmetric cryptographic keys between two nodes in a CAN network in a plug-and-play manner. To this end, special properties of the CAN physical layer are exploited. It can be implemented with no or only minor extensions to CAN controllers and it is particularly suited to enhance the security against remote (and thus scalable) attacks. The generation of group keys between a group of nodes becomes possible if the approach is integrated into a suitable protocol flow.

System and attacker model

In the following, we consider a setup as depicted in Figure 1. Two devices (Alice and Bob) are connected to the same CAN network segment and want to establish a pair of symmetric cryptographic keys. Afterwards, they may use these keys to encrypt and/or authenticate any messages exchanged between them. In addition, there is also a potential attacker (Eve) connected to the same bus segment, which tries to determine or influence the keys to be established between Alice and Bob. We make the following assumptions on Alice, Bob, and Eve:

- All nodes have a similar setup, made up of a CAN transceiver, a suitable CAN controller, as well as a microprocessor running the application software,
- Eve is the victim of a remote attack in the sense that the original software running on that node has been replaced by a modified software,
- Eve can eavesdrop on all messages exchanged on the CAN network. She may also inject arbitrary single bits on the bus by bypassing the CAN controller and directly accessing the CAN transceiver from the malicious software running on the device.

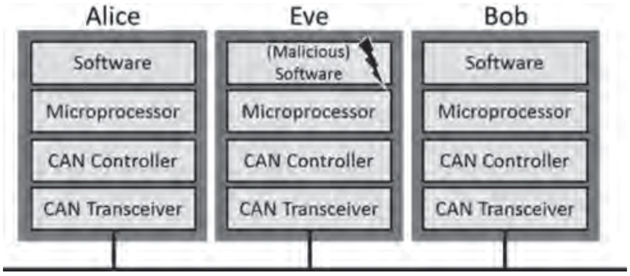


Figure 1: Considered System Model (Photo: Bosch)

It is important to make sure that if one device has been successfully attacked (here: Eve), the impact on the overall system is kept to a minimum. In the attack on a regular car reported in [1], the head unit was successfully compromised first. With proper security mechanisms in place, it would be hardly possible for the head unit to control safety-relevant functions like steering or braking by injecting CAN messages. However, any security mechanism is only secure ▷

as long as the involved cryptographic keys of the legitimate nodes (here: Alice and Bob) remain secret. Therefore, Eve must not be able to determine and/or influence these keys.

Review: Security for CAN networks

The protection of the integrity of a message and the assurance of the authenticity of its sender should be among the top security goals in CAN-based networks. Unauthorized manipulations have to be prevented or should at least be detectable. Encryption, in contrast, makes it harder for an attacker to learn the current system state or becomes necessary for delivering critical software updates.

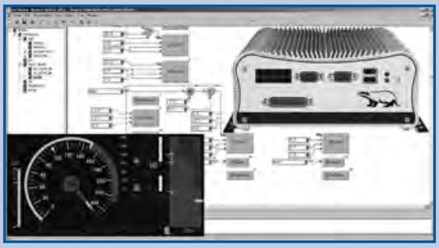
In principle, all these things could be realized in exactly the same way as in the conventional IT world, but for optimal performance the constraints of CAN-based networks must be taken into account. This includes the limited data rate and message sizes, as well as the limited computational power and memory of CAN devices. Therefore, [6] and [7] propose several security mechanisms specifically optimized for CAN. Symmetric cryptographic schemes turn out to be the basis for most of the proposed schemes due to their limited computational complexity and bandwidth requirements. The use of symmetric cryptography requires the availability of symmetric keys at the involved nodes and the distribution and establishment of these keys represents a challenge. Possible options include a manual distribution of keys, e.g., at the end of a production line. This involves a considerable organizational effort and is made invalid if one or several devices were compromised before being integrated into the network. Besides, an automated refreshment of keys cannot be realized this way. An alternative approach is the use of key establishment schemes based on asymmetric cryptography, such as the Diffie-Hellman key exchange protocol [8], [9]. Drawbacks of this approach are the high computational complexity as well as the large amount of data that has to be exchanged in order to set up a secure symmetric key. Moreover, the security of the Diffie-Hellman key exchange relies only on the difficulty to efficiently solve the discrete logarithm problem on finite fields or elliptic curves using state-of-the-art methods. Therefore, once an efficient algorithm for solving these problems has been found, the approach may suddenly become insecure.

We therefore propose a novel approach for establishing symmetric cryptographic keys between two nodes, whose security does not rely on hard mathematical problems, but rather on physical properties of the CAN network. Furthermore, it has a low complexity, low bandwidth requirements, and can be implemented in existing systems. Finally, it may also be used for refreshing established keys.

CAN-based key establishment

The basic idea of our approach is that Alice and Bob agree on a shared secret or key by means of a public discussion using CAN messages. In particular, both nodes simultaneously transmit CAN frames, so that Eve can only see the superposition of both messages. However, since Alice and Bob themselves know what they have transmitted and since they can see the superposition of both messages as well, they may conclude what the other peer has

▷



transmitted and thus establish a shared key that is unknown to Eve.

For the realization, we rely on the property of the CAN network that bit ‘0’ is dominant and bit ‘1’ is recessive, which represents also the basis for the classical bus arbitration. If Alice and Bob simultaneously transmit a certain bit, there are in total four different cases that may occur. These are put together in Table 1. If one of the two nodes transmits a dominant bit (‘0’), the effective bit on the CAN network is also a ‘0’ and only if both nodes transmit a recessive bit (‘1’), we also have the recessive state after the superposition on the CAN network. Therefore, the CAN network may be considered a logical AND function of the individually transmitted bits.

Table 1: Possible combinations of dominant and recessive bit transmissions

Alice	Bob	Effective Bit on CAN Bus
0	0	0
0	1	0
1	0	0
1	1	1

The actual procedure for agreeing on a shared secret between Alice and Bob is a multistep approach as follows:

- Alice and Bob independently generate random bit strings R_{Alice} and R_{Bob} of a predefined length N.
  Example (for N = 10):
  R_{Alice} = 0 1 1 0 1 0 0 1 0 1
  R_{Bob} = 1 0 1 1 0 1 0 1 1 0
- Alice and Bob extend these random bit sequences in such a way that after each bit the corresponding inverse bit is inserted, leading to the modified bit sequences S_{Alice} and S_{Bob} of length *2N*.
  Example:
  S_{Alice} = 01 10 10 01 10 01 01 10 01 10
  S_{Bob} = 10 01 10 10 01 10 01 10 10 01
- Alice and Bob simultaneously transmit the bit sequences S_{Alice} and S_{Bob}, leading to the superimposed bit sequence S_{eff} on the CAN network, which is given as $S_{eff} = S_{Alice}$ AND S_{Bob}.
  Example:
  S_{eff} = 00 00 10 00 00 00 01 10 00 00
  Eve may easily determine this bit sequence as well by means of simple passive eavesdropping on the channel. Therefore, this sequence does not provide any direct advantage yet.
- Alice and Bob determine all tuples in S_{eff} which include a ‘1’.
  Example: In S_{eff} given above, there is a ‘1’ in tuples number 3, 7, and 8 (assuming that we start counting the tuples with 1).
- Alice and Bob delete the bits in their original random bit sequences R_{Alice} and R_{Bob} corresponding to the tuples which included a ‘1’ as determined in step 4. The results are two shortened bit sequences, denoted as K_{Alice} and K_{Bob}.
  Example: Based on the outcome of step 4, Alice and Bob have to delete the bits at positions 3, 7, and 8 in their original bit sequences R_{Alice} and R_{Bob}. We get:
  $K_{Alice} = 0\;1\;\cancel{1}\;0\;1\;0\;\cancel{0}\;\cancel{1}\;0\;1 \quad = 0\;1\;0\;1\;0\;0\;1$
  $K_{Bob} = 1\;0\;\cancel{1}\;1\;0\;1\;\cancel{0}\;\cancel{1}\;1\;0 \quad = 1\;0\;1\;0\;1\;1\;0$
  This is done because whenever the effective bit on the CAN network is a ‘1’, it is clear that both Alice and Bob must have transmitted a ‘1’. Likewise, since the two bits in a tuple are always inverse to each other, it is also clear that in that case both nodes must have transmitted a ‘0’ for the other bit. However, exactly the same conclusion can be drawn by Eve and therefore the tuples including a ‘1’ do not provide any usable information for us as no secrecy is contained. For that reason, these bits are simply removed from S_{eff}.
- The resulting bit sequence of Alice is now exactly the inverse of the corresponding bit sequence of Bob, which is the established shared secret.

What remains after step 5 are the bits that are different in the initial bit strings R_{Alice} and R_{Bob}. When simultaneously transmitting S_{Alice} and S_{Bob}, we always get ‘00’ for the tuples corresponding to these bits. Hence, by eavesdropping on S_{eff}, Eve only knows that Alice and Bob have inverse bits in their original random bit sequences R_{Alice} and R_{Bob} at that position, but she is not able to tell which one has the zero and which one has the one. Alice and Bob, in contrast, know which bit they have transmitted themselves, they can also conclude that the respective other peer has transmitted the inverse bit by evaluating S_{eff} and therefore they have a clear advantage compared to Eve.

With the proposed scheme it is possible to establish a shared secret between Alice and Bob by means of a simple public discussion, i.e., by simply transmitting and receiving CAN frames and interpreting the superimposed frames on the bus. Consequently, the complexity is extremely low, especially compared to existing key establishment schemes and can be done in an automated manner.

In a practical realization, the simultaneous transmission of the bit strings S_{Alice} and S_{Bob} would be done in the payload part of a CAN frame. It is possible to implement the proposed scheme in such a way that other nodes (apart from Alice and Bob) see valid CAN frames and do not trigger the transmission of any error frame. The number of payload bits in a CAN frame is limited to 64 bits in case of Classical CAN and 512 bits in case of CAN FD and the length of the shared secret is not constant, but depends on how many values are equal in R_{Alice} and R_{Bob}. Therefore, the length of the shared secret that can be obtained from one simultaneous message exchange may vary between 0 and *N*, with an expected value of *N/2*. Since the initial random sequences are extended by a factor of two by inserting the inverse bit after each bit and since all these *2N* bits have to be transmitted over the CAN network, the overall efficiency ρ, which relates the length of the usable shared secret after one round of the proposed approach to the number of required bits to establish this shared secret, is given by

$$0 \leq \rho \leq ½,$$

with an expected value of E[ρ] = ¼. This means that on average four payload bits have to be simultaneously transmitted by Alice and Bob in order to establish one secret bit. To achieve state-of-the-art security, symmetric keys of 128 bit or even 256 bit are required, which is why a single run of the proposed approach is not enough to generate a sufficient number of secret bits. Therefore, suitable protocol mechanisms are required, which enable the generation ▷

of keys of arbitrary lengths. This may be done by repeatedly performing the proposed procedure and combining the secret bits generated during each run.

Our approach cannot only be used to generate keys of arbitrary length, but to periodically refresh existing keys. By doing so, certain attacks become more difficult and the potential damage if a particular key is revealed can be limited. Therefore, periodic key refreshment is a highly recommended security practice ([10], [11]). For refreshing a key, a limited number of secret bits is sufficient as they may be combined with the old key. The procedure can be inserted into regular CAN communication in order to generate new secret bits and to refresh the used keys.

Finally, CAN is a multicast-based communication protocol and messages transmitted by one node usually have to be received by multiple nodes. This implies that in many cases not only the communication between two nodes has to be secured, but rather the communication between groups of several nodes. Hence, cryptographic schemes for message authentication, encryption, etc. may only be applied if all devices belonging to a certain group are in possession of the same cryptographic key. The procedure proposed in this paper, however, cannot be extended to a multi-node setup in a straightforward manner. Nevertheless, solutions for establishing group keys are available. In the simplest case, all nodes of a certain communication group could establish a pairwise key with one particular node of that group and then this node may generate a group key and signal it to all nodes of the group, encrypting it with the previously established pair-wise keys.

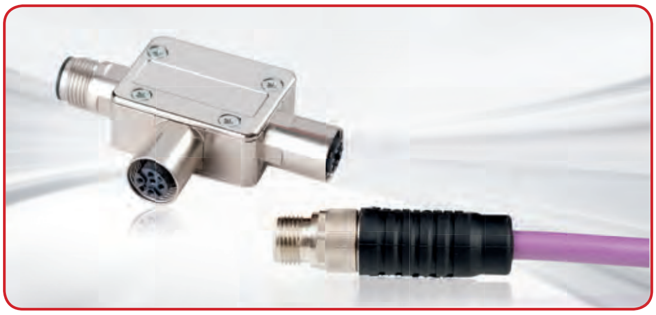
Implementation aspects

As already mentioned, the bit strings S_{Alice} and S_{Bob} should be transmitted in the payload field of a CAN frame. Without any additional measures, this may lead to problems and/or compatibility issues. In a direct implementation, the superimposed CAN frame may violate the bit stuffing rule since even if the individual bit strings S_{Alice} and S_{Bob} adhere to this rule, it cannot be assured that this is also the case for the effective bit string S_{eff}. Other nodes may generate an error frame if they observe such a violation on the bus and clock resynchronization may become more difficult.

A solution to fix this problem is to insert a fixed bit change ('01' or '10') in both S_{Alice} and S_{Bob} after each sequence of at most four bits. This way, Alice and Bob would always transmit the same two bits at this position and since the two bits include a bit change, the bit stuffing rule is never violated in the error-free case. This would come at the cost of a higher overhead. Alternatively, Alice and Bob could determine on-the-fly when it is necessary to insert a stuff bit. Both nodes have to read back the effective bit sequence S_{eff} anyway and could thus check if there have been five identical bits and dynamically insert the inverse bit afterwards. Compared to the first solution, the additional overhead would be lower, but the complexity and processing requirements are higher.

A similar problem occurs with the cyclic redundancy check (CRC) field of a CAN frame in case of a direct implementation of the proposed approach. Since S_{eff} depends on both S_{Alice} and S_{Bob}, the valid value for the CRC field of

▷



the effective message on the bus is generally not equal to the superimposed CRC fields of the messages transmitted by Alice and Bob in case that they calculate the CRC field in such a way that the transmitted frames are valid. In order to solve this issue and assure full backwards compatibility to standard CAN, the correct CRC value that matches the effective CAN frame on the bus could also be calculated by both nodes on the fly and then be appended to that frame after the payload field. While making sure that the effective frame on the bus is a valid CAN frame (based on existing specifications), it has the nice side effect that this procedure automatically helps to make sure that both Alice and Bob have received the same effective bit string S_{eff} (which is essential for deriving the same shared secret). If the two nodes are receiving differing bits, they would append different CRC fields, which would be detected by one of the nodes if a recessive bit was overwritten by a dominant one.

With these approaches to dealing with bit stuffing violations and the CRC field, it is possible to achieve full backward compatibility in the sense that all frames on the CAN network are in line with the existing specifications – at least in the error-free case. Smooth migration paths become possible, where not all nodes connected to a CAN network necessarily have to support the proposed approach and existing hardware and software can be reused. We envision flexible implementation options, where existing CAN controllers do not have to be modified as long as they are supplemented by an additional hardware or software module.

Security considerations

If Eve is only passively eavesdropping on the CAN network, she will not be able to readily determine the established shared secret bit sequences K_{Alice} or K_{Bob}: she only knows that the remaining bits were different for both nodes, but cannot tell who has transmitted the zero and who the one. If, in contrast, Eve is trying to perform an active attack, for example by sending additional own bits during the exchange of S_{Alice} and S_{Bob}, two different possibilities have to be considered:

- Eve transmits a recessive bit,
- Eve transmits a dominant bit.

Transmitting a recessive bit is no different from not transmitting at all since a recessive bit does not change the effective state on the CAN network. Therefore, we only have to analyze what Eve might does by superimposing another dominant bit to the bits exchanged between Alice and Bob. To this end, it is important to remember that with our procedure only those bits remain in the final shared secret for which the effective bit on the CAN network was ‘0’ for both the transmission of the original and the inverse bit. Moreover, if Eve transmits a dominant bit, she cannot tell what the status on the CAN network would have been without her transmission. Therefore, we can conclude the following:

Conclusion 1: An active Eve may disturb our procedure in such a way that the generated secret bit strings K_{Alice} and K_{Bob} are actually not equal on both sides. In order to be able to detect such cases, additional mechanisms should be introduced, with which Alice and Bob can verify that they have generated the same bit sequences. This could be done by calculating and exchanging a hash value of these bit sequences.

Conclusion 2: An active Eve is not able to enforce the generation of a particular shared secret between Alice and Bob (which she then would be aware of) and/or to learn the shared secret that is established between both nodes. This is because K_{Alice} and K_{Bob} depend not only on S_{eff} (which may be determined and influenced by Eve), but also on the bits of R_{Alice} and R_{Bob}, which are unequal on both sides, and Eve has no way to determine these.

Conclusion 3: An active Eve may perform a denial-of-service attack by preventing the establishment of a shared secret, for example by continuously sending a frame with the highest priority. However, this threat exists basically for any scheme since an active Eve could block any CAN communication on the bus. In this case, the failsafe mode of all devices should prevent any serious safety-critical impact.

Conclusion and way forward

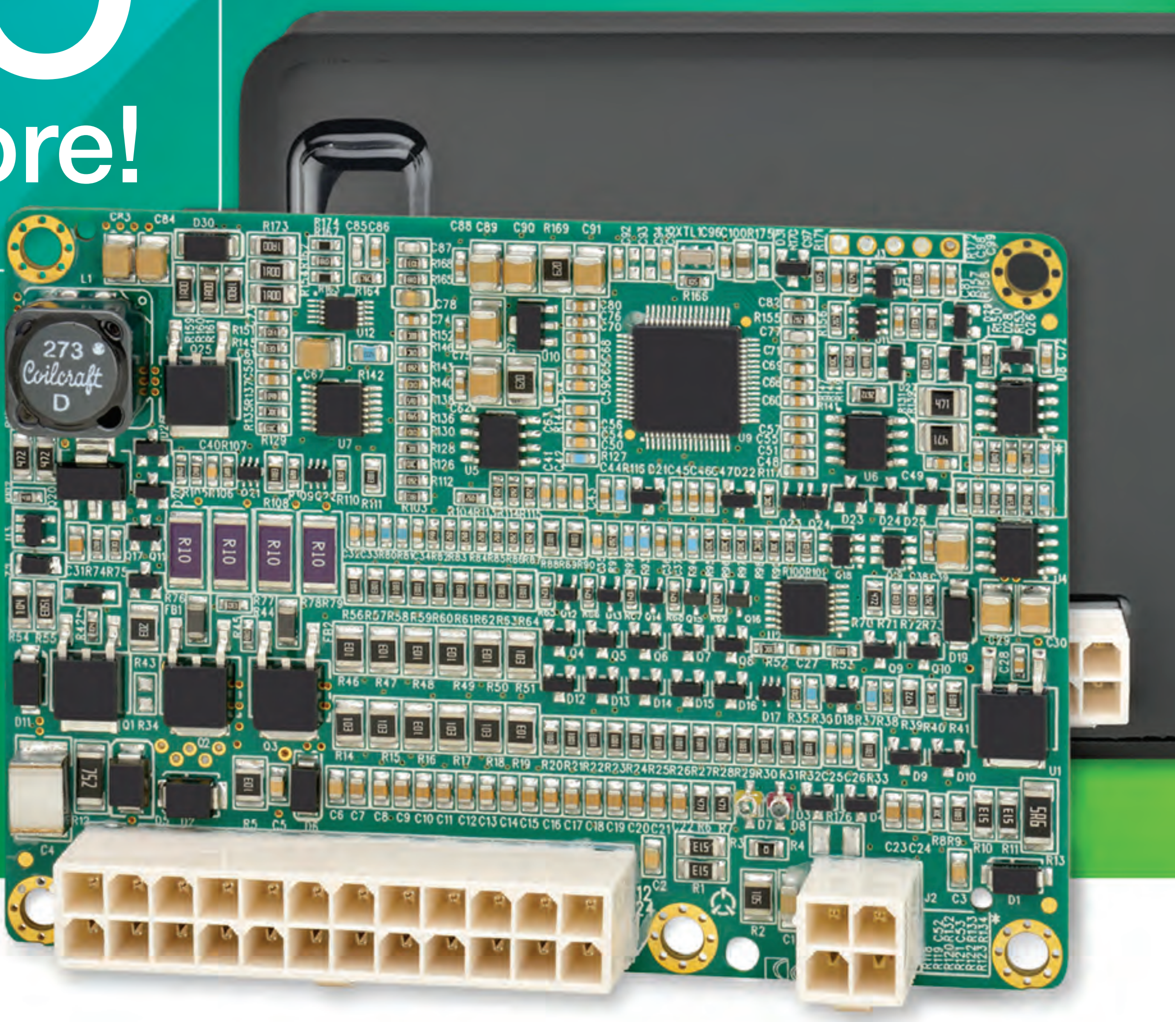
Security will play a crucial role for the success and acceptance of connected systems. A challenge in this regard is how to distribute and manage the cryptographic keys between the involved nodes. We have proposed a novel approach to establishing and/or refreshing symmetric cryptographic keys between two CAN devices in a plug-and-play manner, exploiting special properties of the CAN network. The proposed scheme requires only the simultaneous exchange of random bit sequences along with an appropriate interpretation of the resulting effective bit sequence on the bus. Therefore, it is of very low complexity and may be readily implemented and integrated in practical systems. Even though it cannot address all existing security challenges, it has the potential to become a major building block for secure CAN communication in future. Also, it should be noted that exactly the same concept may also be used in conjunction with other bus systems having similar properties as CAN. Apart from all CAN derivatives, such as TTCAN or CAN FD, this includes the LIN- and I²C-bus.

As a next step, the basic idea has to be embedded in a larger framework, including suitable protocols and mechanisms for synchronized frame transmissions between Alice and Bob, the establishment of group keys, the generation of keys of arbitrary lengths and the like. A first practical proof-of-concept demonstration is planned. In general, no major showstoppers are expected in this respect. ◀

Authors

Andreas Mueller and Timo Lothspeich
Robert Bosch GmbH
www.bosch.com
Andreas.Mueller21@de.bosch.com
Timo.Lothspeich@de.bosch.com

I/O Galore!



THE CURTIS MODEL 1356 CANbus I/O expansion module adds digital, analog and encoder I/O to any CANopen based control system and is ideal for industrial and manufacturing equipment, engine powered and electric vehicles. This low cost, powerful, compact general purpose I/O module, 100 x 70 mm, is available as a PCB or encapsulated. Two models at 12 V-80 V, provide 18 multi- purpose I/O pins with: 13 digital and 5 analog inputs; 1 and 3 Amp outputs configurable as PWM, Constant Current or Constant Voltage; 1 quadrature input; a serial port; 12VDC unregulated and a 5VDC regulated power supplies. The encoder input is suited to reading quadrature encoders from steer or positioning devices with automatic conversion to speed, RPM and distance travelled. The 1356 module is programmed via serial or CANopen for simple integration.

See for yourself. Contact Curtis to expand your I/O.

www.curtisinstruments.com

To see product information and datasheet go to: **www.bit.do/Curtis1356**

1960 · 2015 55 YEARS CURTIS

CURTIS

Structured software development

Codesys has extended IEC 61131-3 application engineering by reducing the development effort that is caused by recurring procedures: The Codesys Development System provides automated methods.

For certain mobile machines, it is mandatory to fulfill standard compliances according to IEC 61508 for SIL-2 or SIL-3 safety controllers and their software components, such as applications and communication-stacks. Traditionally, engineering requires many manual tasks to comply with the standard. Modern engineering already benefits from certified communication stacks, for example CANopen Safety.

For safety development, specific methods are recommended for the development of life-cycle models. To reduce the risk of expensive field service operations, the main focus of the methods must be on coding, debugging, and testing, followed by concept work and administrative tasks, such as code management. With automated methods fully integrated into the Codesys Development System, the following recommended IEC 61508 methods can be completed more efficiently and reliably – regardless of the chosen development model.

The Unified Modeling Language (UML) was invented to graphically describe different aspects of the application software. In Codesys, it is possible to use UML in each IEC 61131-3 project as an additional integrated programming language, supporting two types of engineering. The class diagram is a type that shows the internal software structure by means of interfaces and software classes. The state chart is used to describe the behavior of the software during different states. UML state charts are similar to SFC diagrams and can be used to structure the different logical application steps.

Codesys UML is an ideal add-on for object-oriented application programming. Furthermore, it provides an efficient way for communicating with customers and developers by means of a single tool. It supports continuous engineering in IEC 61131-3 languages based on information that is part of the UML diagram by reusing UML models in the application.

The code administration is a non-functional method for improving the code quality of safety software by means of prevention. It documents the meta-information. Codesys SVN extends the capabilities of the Codesys Development System by interfacing an Apache Subversion server, so that the single objects that are part of the project file can be administrated. Thus different developers can work on dedicated objects of the same project at the same time. Furthermore, branches of the developed project can be administered and versioned.

The execution of checklists and formal inspections are a minimum recommendation for SIL-1 to SIL-4 safety levels, and formal inspections are even highly recommended.

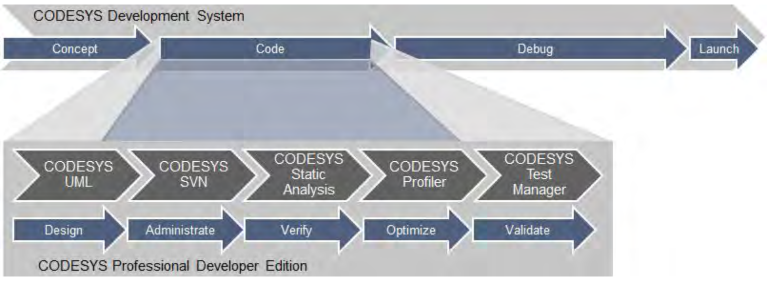


Figure 1: Methodical engineering support (Photo: 3S)

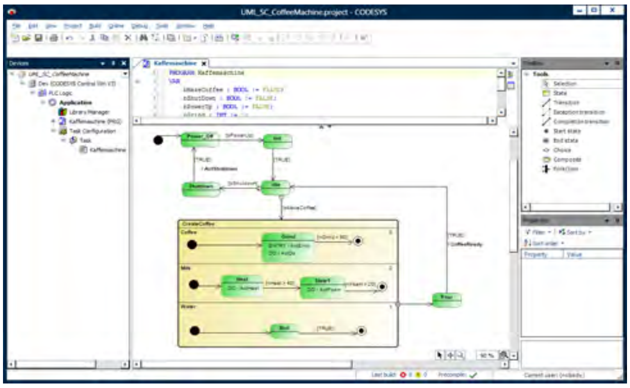
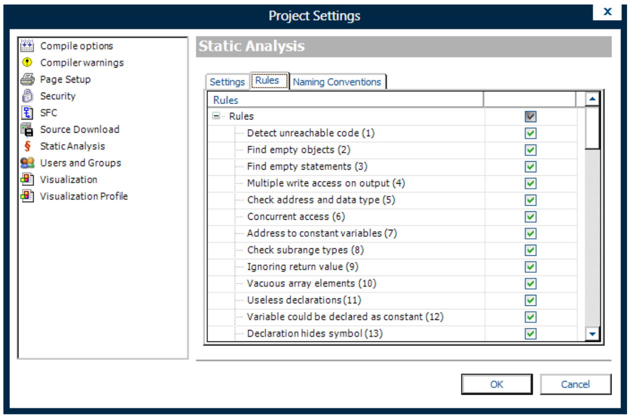


Figure 2: Architectual design with Codesys UML (Photo: 3S)



Project Settings

- Compile options
- Compiler warnings
- Page Setup
- Security
- SFC
- Source Download
- Static Analysis
- Users and Groups
- Visualization
- Visualization Profile

Static Analysis

Settings | Rules | Naming Conventions

Rules	
Rules	☑
Detect unreachable code (1)	☑
Find empty objects (2)	☑
Find empty statements (3)	☑
Multiple write access on output (4)	☑
Check address and data type (5)	☑
Concurrent access (6)	☑
Address to constant variables (7)	☑
Check subrange types (8)	☑
Ignoring return value (9)	☑
Vacuous array elements (10)	☑
Useless declarations (11)	☑
Variable could be declared as constant (12)	☑
Declaration hides symbol (13)	☑

OK | Cancel

Figure 3: Automated quality assessment with Codesys Static Analysis (Photo: 3S)

for SIL-3 and SIL-4. PLCopen recognized these requirements and provides a technical paper with coding guidelines for IEC 61131-3 applications as of July 2015.

The primary aim of Codesys Static Analysis is the improvement of the quality of the application code by automatic detection of code weaknesses and anomalies. Using this tool, the manual effort of code reviews and metrics calculations can be reduced. Codesys Static ▷

CAN FOR EXTREME ENVIRONMENTS

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

750-338/040-000

750-8204

750-837

750-838/040-000

750-337

750-338

750-347

750-348

767-1501

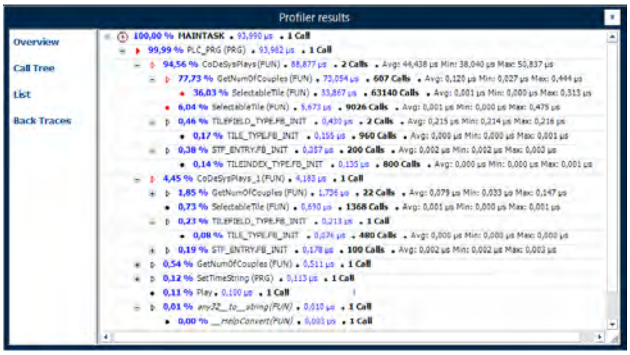
767-658 (CAN)

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

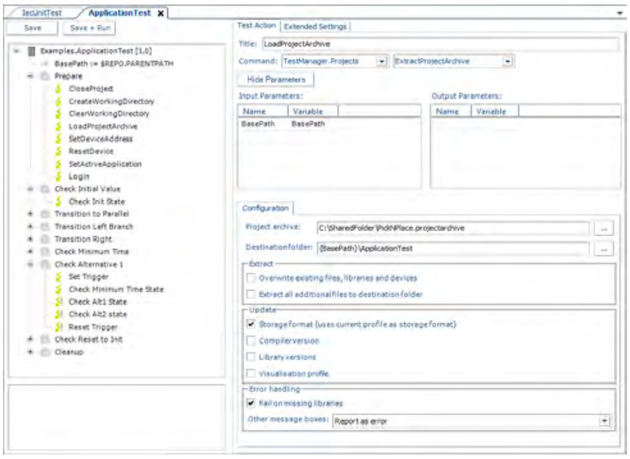
www.wago.com

WE INNOVATE!



```
100,00 % MAINTASK • 93,990 µs • 1 Call
  99,99 % PLC_PRG (PRG) • 93,982 µs • 1 Call
    94,56 % CoDeSysPlays (FUN) • 88,877 µs • 2 Calls • Avg: 44,438 µs Min: 38,040 µs Max: 50,837 µs
      77,73 % GetNumOfCouples (FUN) • 73,054 µs • 607 Calls • Avg: 0,120 µs Min: 0,027 µs Max: 0,444 µs
        36,03 % SelectableTile (FUN) • 33,867 µs • 63140 Calls • Avg: 0,001 µs Min: 0,000 µs Max: 0,313 µs
      6,04 % SelectableTile (FUN) • 5,673 µs • 9026 Calls • Avg: 0,001 µs Min: 0,000 µs Max: 0,475 µs
      0,46 % TILEFIELD_TYPE.FB_INIT • 0,430 µs • 2 Calls • Avg: 0,215 µs Min: 0,214 µs Max: 0,216 µs
        0,17 % TILE_TYPE.FB_INIT • 0,155 µs • 960 Calls • Avg: 0,000 µs Min: 0,000 µs Max: 0,001 µs
      0,38 % STF_ENTRY.FB_INIT • 0,357 µs • 200 Calls • Avg: 0,002 µs Min: 0,002 µs Max: 0,003 µs
        0,14 % TILEINDEX_TYPE.FB_INIT • 0,135 µs • 800 Calls • Avg: 0,000 µs Min: 0,000 µs Max: 0,001 µs
    4,45 % CoDeSysPlays_1 (FUN) • 4,183 µs • 1 Call
      1,85 % GetNumOfCouples (FUN) • 1,736 µs • 22 Calls • Avg: 0,079 µs Min: 0,033 µs Max: 0,147 µs
      0,73 % SelectableTile (FUN) • 0,690 µs • 1368 Calls • Avg: 0,001 µs Min: 0,000 µs Max: 0,001 µs
      0,23 % TILEFIELD_TYPE.FB_INIT • 0,213 µs • 1 Call
        0,08 % TILE_TYPE.FB_INIT • 0,076 µs • 480 Calls • Avg: 0,000 µs Min: 0,000 µs Max: 0,000 µs
      0,19 % STF_ENTRY.FB_INIT • 0,178 µs • 100 Calls • Avg: 0,002 µs Min: 0,002 µs Max: 0,003 µs
    0,54 % GetNumOfCouples (FUN) • 0,511 µs • 1 Call
    0,12 % SetTimeString (PRG) • 0,113 µs • 1 Call
    0,11 % Play • 0,100 µs • 1 Call
    0,01 % any32__to__string(FUN) • 0,010 µs • 1 Call
      0,00 % __HelpConvert(FUN) • 0,005 µs • 1 Call
```

Figure 4: Time analysis on the device with Codesys Profiler (Photo: 3S)



```
Examples.ApplicationTest [1.0]
  BasePath := $REPO.PARENTPATH
  Prepare
    CloseProject
    CreateWorkingDirectory
    ClearWorkingDirectory
    LoadProjectArchive
    SetDeviceAddress
    ResetDevice
    SetActiveApplication
    Login
  Check Initial Value
    Check Init State
  Transition to Parallel
  Transition Left Branch
  Transition Right
  Check Minimum Time
  Check Alternative 1
    Set Trigger
    Check Minimum Time State
    Check Alt1 State
    Check Alt2 state
    Reset Trigger
  Check Reset to Init
  Cleanup
```

Title: LoadProjectArchive

Command: TestManager.Projects | ExtractProjectArchive

Input Parameters:

Name	Variable
BasePath	BasePath

Output Parameters:

Name	Variable

Project archive: C:\SharedFolder\PickNPlace.projectarchive

Destinationfolder: {BasePath}\ApplicationTest

Extract

☐ Overwrite existing files, libraries and devices

☐ Extract all additionalfiles to destination folder

Update

☑ Storage format (uses current profile as storage format)

☐ Compilerversion

☐ Library versions

☐ Visualisation profile

Error handling

☑ Fail on missing libraries

Other message boxes: Report as error

Figure 5: Automated testing with Codesys Test Manager (Photo: 3S)

Analysis checks the application code against numerous integrated coding guidelines and naming conventions. Furthermore, it calculates metrics of the IEC 61131-3 program. For a maximum benefit, the tool is integrated seamlessly into the IEC 61131-3 IDE so that application code import or export is not required.

Time measurement on the controller can be provided by event-based measurement tools, such as Codesys Profiler. The runtime between two predefined code points is measured by instrumentation of the code in order to fulfill the recommendations for SIL-2 to SIL-4. Codesys Profiler automatically adds and removes extra measurement code at the calling and exit points of the POU (Program Organization Unit). As a result, the following information can be retrieved from the controller:

- Total time spent in call,
- Average time of all POU calls of a single cycle,
- Minimum and maximum processing time over multiple cycles,
- Number of calls,
- Standard deviation of average measured time.

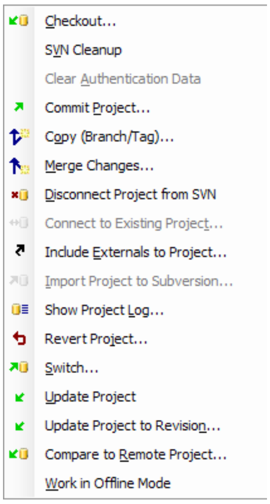

Checkout...
SVN Cleanup
Clear Authentication Data
Commit Project...
Copy (Branch/Tag)...
Merge Changes...
Disconnect Project from SVN
Connect to Existing Project...
Include Externals to Project...
Import Project to Subversion...
Show Project Log...
Revert Project...
Switch...
Update Project
Update Project to Revision...
Compare to Remote Project...
Work in Offline Mode

Figure 6: Seamlessly integrated project administration with Codesys SVN (Photo: 3S)

After profiling is switched off, the extra measurement code is deleted automatically, thus no unused code exists in the final application.

Safety-compliant software testing has to include independent test specifications, confirmation of intended functions, and test documentation. The Codesys Test Manager provides two options for performing tests and fulfilling IEC 61508 requirements. Tests can be written in IEC 61131-3 for execution in real time on the controller or they can be defined to operate via the monitoring interface. The tests can be developed in parallel to the application. The planned test cases can be described directly from the integrated editor and referenced to the appropriate requirement. When the tests have been performed on the device, the results are displayed in a result window and can be saved to a file or directly within the Codesys Development System. With continuous model-based testing, the functional quality is always transparent and can be documented properly. Furthermore, the effort for recurring tests is reduced considerably.

Development tools which provide methodical development support make engineering life easier. They reduce the manual effort for safety certifications and improve the quality of IEC 61131-3 applications in automation technology. In addition, non-safety applications also benefit from conventional methods known throughout the IT world. Although they are not mandatory, these methods help reduce unwanted application behavior, field service missions and increase application quality right away. ◀

Author

Michael Schwarz
3S-Smart Software Solutions GmbH
www.codesys.com

Automatic leveling system for graders

In no other industry, competitive stress is as severe as in the construction industry. Economic efficiency needs to be constantly improved through increased working speed and improved quality.

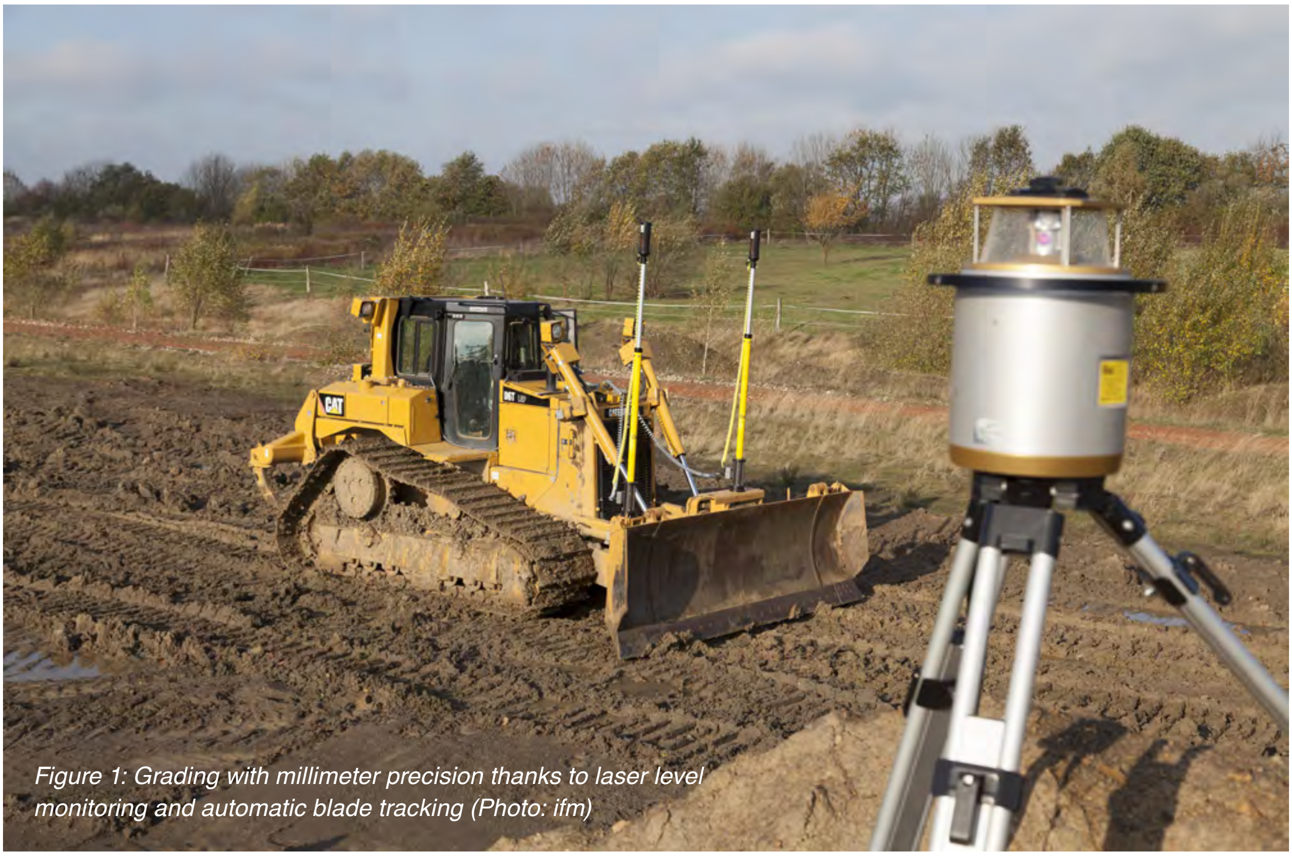


Figure 1: Grading with millimeter precision thanks to laser level monitoring and automatic blade tracking (Photo: ifm)

The GRi-P1 Gritzke automatic leveling system for graders increases the flexibility and productivity of the machines significantly. This helps save material costs for earth moving and fine grading. The system can be equipped with several sensors and it combines easy handling with a self-explanatory user interface. Each millimeter counts when miles of road sections under construction need to be graded to the same level. A grading that is only one millimeter higher than required can easily cause several truckloads of additional material.

Laser leveling

Leveling long stretches with millimeter precision is only possible by applying efficient modern technology. Here, laser-based systems have proven to be particularly accurate, cost-efficient, and reliable. Function principle: A laser fixed to a tripod rotates around its own axis to create a laser level. This level can be adjusted in parallel to the required surface. A vertical photoelectric receptor cell mounted to the grader blade receives the laser beam.

An intelligent controller tracks the laser receptor and the grader blade to make sure they are always at the exact height with the laser projection level. This way, the driver can focus on the horizontal movements of the

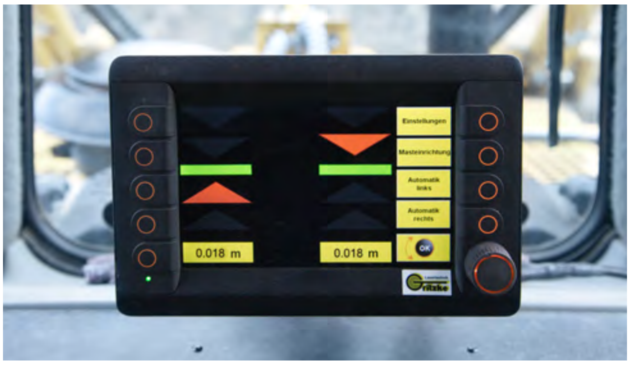


Figure 2: Working surface with following setting options: "Mast installation" e.g. "search laser" or "park mast", automatic mode for the right or the left mast "on / off", "mast adjustment on both sides", as well as current height indication in 1/10 mm (Photo: ifm)

▷

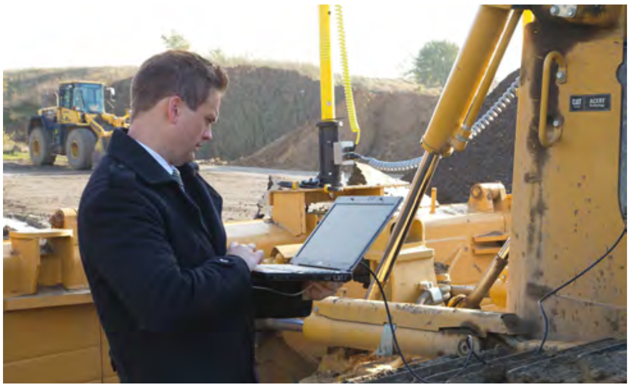


Figure 3: Dennis Blume from ifm on site: The controller was tested on the machine and the programming adjusted again and again (Photo: ifm)

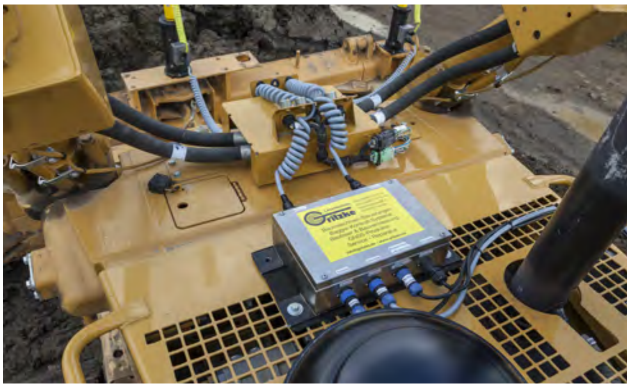


Figure 4: Gritzke developed and built the marketable system in cooperation with ifm Electronic (Photo: ifm)

grader while the blade is automatically kept at reference height with millimeter precision. To level sloped surfaces, the laser can simply be adjusted in parallel to the required slope. Depending on whether the grader movements are longitudinal, transverse, or diagonal to the slope, different lateral blade inclinations are required. With laser-based systems, the blade inclination can be controlled automatically. For this purpose, a second laser receiver is installed on one side of the grader blade. Alternatively, an inclination sensor and/or an ultrasonic sensor is used on the blade.

Gritzke Lasertechnik OHG is based in Lemgo, Eastern Westphalia. It is specialized in the development, production, and distribution of construction machine control and positioning systems. One of its central credos is: Circumvent the product monopoly constraints of the market leaders. For this reason, the systems can be installed on any machine, even if it is already pre-equipped with cables.

To distinguish themselves from common systems, Gritzke has considered the advantages and disadvantages of all systems while putting their own ideas into practice.

A new flexible system

In the past, Gritzke used programmed controllers from different manufacturers to control the leveling systems. A disadvantage was that the company could not carry out customer and machine-specific adjustments or software modifications as the system integrator. The hardware manufacturers had the ownership of the software. Individual adjustments or modifications were very time-consuming and cost-intensive, or were refused.

Rolf Oschatz, managing director at Gritzke, explained: "About two years ago, I decided that we would develop our own laser-based leveling system for construction machines. The aim was to offer our customers a combination of special user-friendliness, high accuracy, and best-possible competitive price. With our development, we basically did not reinvent the wheel, but combined all advantages of the competitive systems with our ideas and requirements.“

Figure 5: The heart of the system is a 32-bit controller from ifm Electronic for mobile applications (Photo: ifm)

Application know-how is one thing, but when it came to the heart of the system, the controller and its software, Gritzke found their present partner, ifm Electronic, more or less by accident. Rolf Oschatz added: "The development together with the earlier hardware suppliers was rather slow. The initially promised support was very hesitant and consisted of target figures rather than technical support. In April 2013, at ifm Electronic's booth at Bauma, I was asked in an informative conversation if we needed help. Mutual interest arose quickly. What impressed me in particular: They did not ask about possible quantities but promised comprehensive project support." This was the beginning of the close partnership between Gritzke and ifm Electronic. In cooperation with the automation specialist, Gritzke developed, built, and sold the first German CAN-based GRi-P1 leveling system for graders.

The following months were characterized by intensive co-operation. Dennis Blume, sales specialist for control technology at ifm Electronic, had the lion's share in supporting the project. This was done in close cooperation with Gritzke because one important requirement on the new system was to have Gritzke's in-house software know-how. The heart of the installation is the CR0033 CAN-compatible ifm controller for mobile applications. Ifm's CR1084 display with graphics capabilities is used as the operating unit.

Rolf Oschatz explained: "The cooperation with ifm was passionate and successful. Often, we tested the software and hardware outdoors on the machines till late at night. Many thanks in this respect to the company Stork Tongruben und Transportunternehmen in Hiddenhausen who provided us with a caterpillar (Cat D6T) and the site, a clay pit, for thorough testing. And it paid off: after 18 months, we could implement the system until it was ready for the market. Without Mr Blume's exceptional personal commitment, we would never have achieved this in such a short time".

Flexible in the application

The leveling system is the first of its kind to be developed, programmed, and built by only one supplier. The customer benefits from the fact that adaptations, special customer requests, or improvements can be implemented rapidly. Due to its modular concept, the GRi-P1 Gritzke system can be used for any kind of laser-based height monitoring and control for different applications as well as for construction machines.

That means that it can also be used on excavators for depth monitoring, on height and/or pivoting angle limitation, on wheel loader leveling systems, on piling and drilling rigs, on agricultural machines or on container lifts. The customer does not need expensive software updates, since the different applications are already stored and selectable in one software program. The controller can also be used on different machines, if required. The customer saves the purchase of double components such as operating unit, central processing unit, or sensors.

By integrating selectable application programs in a modular device, the development and hardware costs were reduced to a minimum. The result: The Gritzke system costs about one third less than common systems.

High speed - thanks to CAN

It is the first system to use the CAN interfaces for data transfer. The data can be transmitted up to five times faster from the laser receiver or from the inclination sensor/ultrasonic sensor to the controller. This fast data transmission and processing in the controller is necessary to ensure a fast signal chain from the laser receivers to the controller to the valve control of the caterpillar blade. This is the only way to enable work with millimeter precision, even at high speeds. Also the joysticks, switches, and buttons of the construction machine are polled and transmitted via CAN to the process control.

If necessary, the user can manually alter the automatic zero adjustment on the graphic operating unit. Switching – for example to inclination sensor or ultrasonic sensor (for grading according to a ground reference, e.g. curb) – can easily be done on the operating unit.

The heart: the controller

A 32-bit controller from ifm Electronic is the brain of the control system. It has up to 16 multifunctional inputs and outputs as well as four CAN interfaces. The heart of the controller is a modern and fast 32-bit processor integrat-

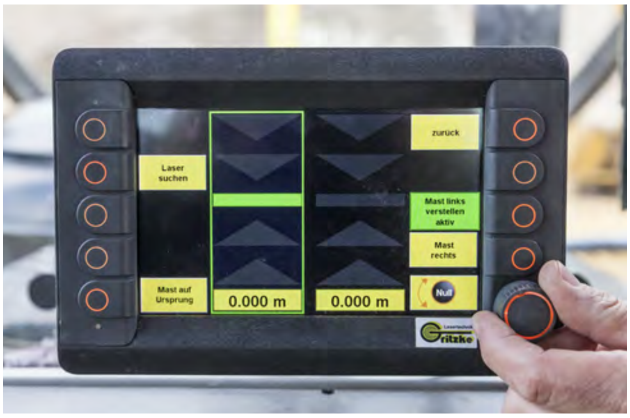


Figure 6: The CR1084 dialog monitor from ifm Electronic for visualization and data entry (Photo: ifm)

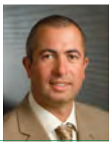
▷

ed into a compact IP67 metal housing. Its monitoring and protective functions enable reliable operation even under extreme operating conditions. The high number of multifunctional inputs and outputs allows adjustment to the respective application using application software (IEC 61131-3 with Codesys). Depending on the type of input, a configuration as digital, frequency, or analog input with diagnostic function or as input for resistance measurement is possible.

The four CAN interfaces are in accordance with ISO 11898 and support various bus protocols and different bit-rates as well as transparent or preprocessed data exchange. The controllers were specially designed for robust applications in vehicles and for mobile automation and can carry out complex and proportional functions reliably.

Thanks to the closed die-cast aluminum housing with its protection rating IP67, the PDM360 NG dialog module can be used outside and inside the cabin – by means of surface or panel mounting. The scratch-resistant 7-inch TFT color display with a resolution of 800 pixels x 480 pixels and a color depth of 18 bits provides brilliant graphical representation. For operation, the module has nine backlit function keys with tactile feedback. In addition, an encoder with pushbutton or a navigation key is available depending on the model. The 32-bit controller is programmable with Codesys according to IEC 61131-3. In addition to the internal 1 GiB memory, the user can connect external media to the integrated USB 2.0 port.

Four CAN interfaces (ISO 11898) support CANopen, J1939, or a free protocol. Together with a 100 Mbit Ethernet interface and the Linux operating system, a universal platform for networking and communication with other vehicle components is formed. Connection is made via robust and safe M12 connections. The project benefits from the application know-how of many years, powerful hardware and, above all, the will to bring about something special together. ◀



Author

Andreas Biniasch
ifm Electronic
www.ifm.com
info@ifm.com

Long-Ma: The birth of a hybrid giant

Long-Ma, the dragon-scaled winged horse of Chinese mythology, came to life during the celebrations of the 50th anniversary of diplomatic relations between China and France. Dintec managed to power the dragon-horse in only 12 months.

Long Ma Jing Shen, an original creation of François Delaroziere and Compagnie La Machine (Photos: ©Stephan Muntaner, all rights reserved)

In October 2013, Compagnie La Machine approached their partner Dintec with the request to power a giant dragon-horse. It should allegorize Long-Ma, a fabled winged horse with dragon scales in Chinese mythology. Very quickly it became obvious that this project would be like no other. As Long-Ma should come to life within only 12 months and then be the main attraction at the celebrations for the 50th anniversary of the diplomatic relations between China and France, the pressure on Dintec was high. But being an experienced system integrator especially for off-highway applications, Dintec took on the challenge and the responsibility for the whole power system. Passion for mobile machines and for the entertainment created by the amazing creatures of Compagnie La Machine allowed Dintec to master the system from power generation, storage, and distribution to control and sensors. A multidisciplinary team of five engineers started by understanding the energy balance of the animal and the constraints in terms of operating mode (hybrid, fully electric for 45 minutes). After 1800 hours of engineering work, a hybrid architecture was defined in detail (shown in Figure 1) and the software for the power management written and compiled. Only seven months after kick-off, a prototype of the system could be seen working in Dintec's workshop.

Hybrid power pack with noise insulation

In order to supply the 620 V_{DC} electric net, Dintec designed a bespoke power pack driven by a Perkins 6-cylinder Tier 4 Interim engine to lower emissions. The diesel engine is coupled to a STW Power Mela 140 kW E-machine. Due to its integrated inverter, the Power Mela permits four-quadrant operation and works as a generator. Running at variable speeds depending on power demand, the combination offers best-in-class fuel consumption. Lots of effort was also made in terms of power pack enclosure, as the giant Long-Ma had to remain a silent machine (see Figure 2). Even when the diesel engine starts, advanced noise insulation keeps Long-Ma quiet.

The main electric consumers are the two 140 kW E-machines working as motors. They supply hydraulic ▷

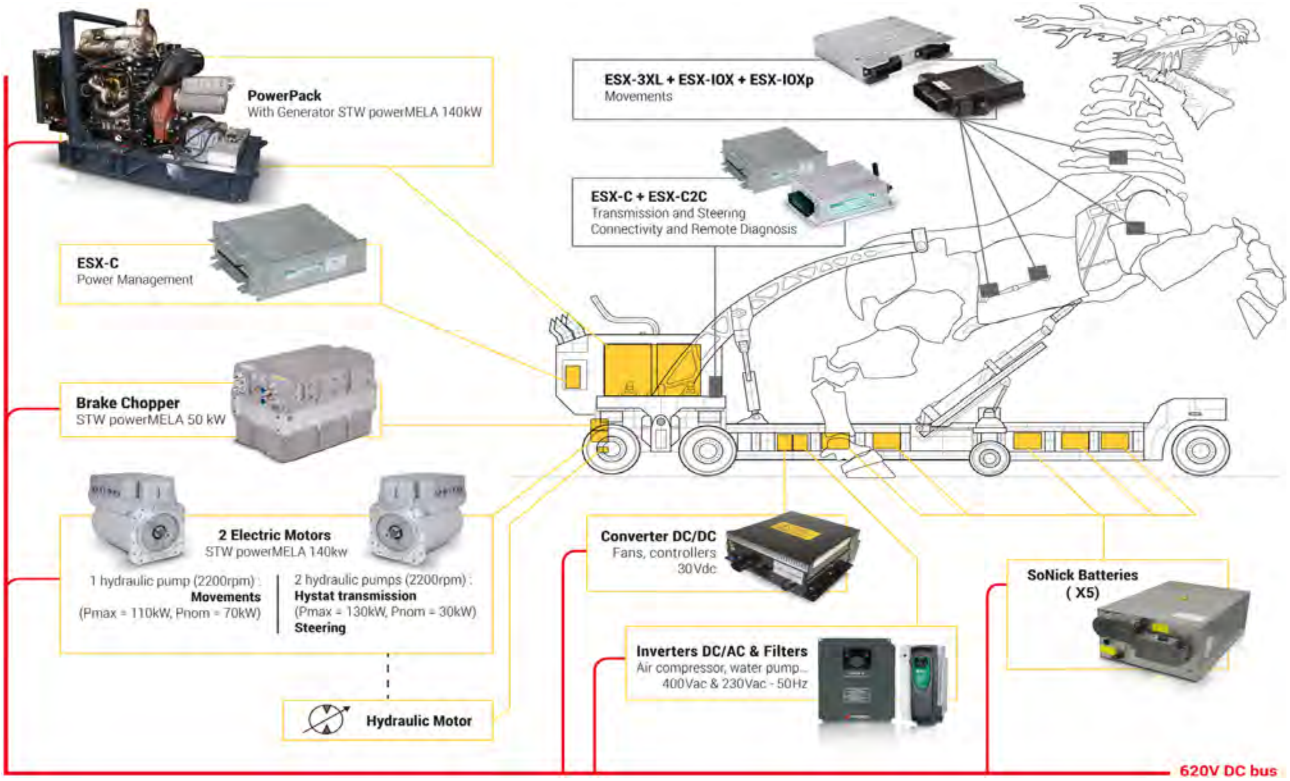

PowerPack
With Generator STW powerMELA 140kW
ESX-3XL + ESX-IOX + ESX-IOXp
Movements
ESX-C + ESX-C2C
Transmission and Steering
Connectivity and Remote Diagnosis
ESX-C
Power Management
Brake Chopper
STW powerMELA 50 kW
2 Electric Motors
STW powerMELA 140kw
1 hydraulic pump (2200rpm) :
Movements
(Pmax = 110kW, Pnom = 70kW)
2 hydraulic pumps (2200rpm) :
Hystat transmission
(Pmax = 130kW, Pnom = 30kW)
Steering
Hydraulic Motor
Converter DC/DC
Fans, controllers
30Vdc
Inverters DC/AC & Filters
Air compressor, water pump...
400Vac & 230Vac - 50Hz
SoNick Batteries
(X5)
620V DC bus

Figure 1: Visual schema of Long-Ma (Photo: Dintec)

power for all movement (head, body, legs, eyes), the hydrostatic transmission, and the steering system. A brake chopper has been implemented on the DC bus to avoid over-voltage in case of hard braking. The brake chopper is also equipped with an electric insulation measurement system that permanently controls the electric isolation of the DC bus from the machine chassis. A shortcut here would be hazardous to the operators of Long-Ma. All electric components (batteries, motors, generator, inverters) are connected to the DC bus through small cabinets integrated into the machine.

The giant also requests a number of additional electric networks, so Dintec supplied appropriate solutions derived from the 620 V_{DC} bus. Inverters and filters take care of the transformation to the AC world with a 400 V_{AC} - 50 Hz (triphases + neutral) net for air compressors and water pumps and a 230 V_{AC} - 50 Hz net to power small AC motors, lights, and music systems. A DC/DC converter converts the voltage down to 30 V_{DC} to supply fans, controllers, and lights.

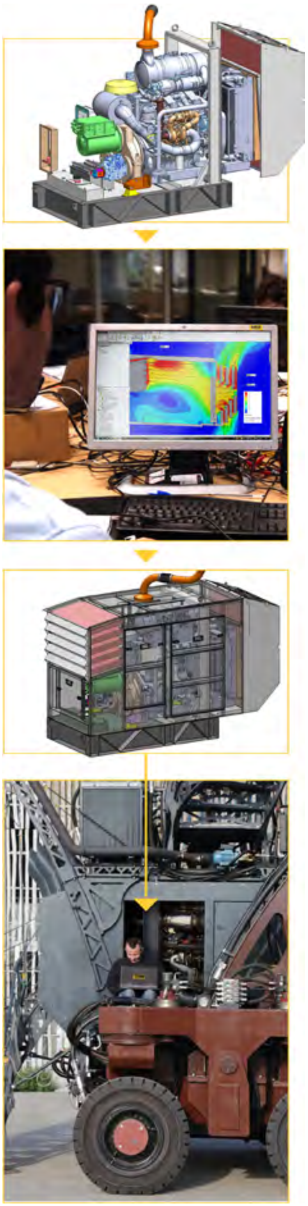


Figure 2: Power Pack design and manufacturing (Photo: Dintec)

Energy storage and electronics

After a deep benchmarking of technologies, taking the duty cycle of the machine, its required lifetime, and the constraints in terms of use and availability into account, Dintec selected Sodium Nickel batteries for their high power and energy density and supplied five sets of 23,56 kWh each fitting to the 620 V_{DC} net and weighing a total of 1000 kg. While each set features its own cooling system, the batteries are designed to be charged from the grid but can also be directly charged from the embedded power pack when necessary.

Animating such an animal smoothly and realistically means over 50 speed and position regulations for the movements of the head, the body, the legs, and the displacement (hydraulic propelling and steering). To handle these numerous functions and the communication between the 24 HMIs (display, joysticks, keypads) and more than 200 sensors (angles, length for cylinder heads, pressures, temperatures, etc.), a complex network of 16 embedded controllers communicating through 17 different CAN networks was defined.

An ESX-3XL control unit was selected as the center and brain of the system. It can handle up to 124 I/Os, has four independent CAN interfaces, and offers the computing power and memory to run sophisticated applications. Steering and propelling are managed through the off-the-shelf STW ESX-C, which is capable of controlling the rolling chassis. Another ESX-C control unit with two CAN interfaces was used to implement the power management algorithms. These are capable of permanently adjusting the power generation to the machine requirements and make sure all requested movements are operated at the right speed. Two ESX-IOX and eleven ESX-IOXp decentral I/O control systems were deployed for the hydraulic movement control of the head, legs, and eyes. Efficient communication between all these CAN participants was established. Lastly, an ESX-C2C with mobile communication capabilities was added so that diagnosis, software updates, and machine follow-up could be done remotely using Dintec's web server.

Finding its home in China

After intensive testing in Nantes, France, Long-Ma passed all controls and certifications handled by TUV Belgium, which is specialized in unusual and therefore special applications. Long-Ma then left Nantes in September 2014 aboard an Antonov 124 to land the next day in Beijing, China. After being re-assembled and after several days of testing and rehearsals for the ceremony, Long-Ma performed three days of intensive shows, leaving the Chinese public with unforgettable memories. Finally, Long-Ma returned back to Nantes in June 2015 for a couple of presentations and performances. After those, Long-Ma will permanently stay in China as part of a larger project.

Both companies – Dintec as the System Integrator and STW as the provider of the controllers, the electric generator and motors – were proud to work on a project of this scale. This pride reached its peak when after 12 months the engineered system gave life to the animal which took part in the big show for the celebrations of the 50th anniversary of China's and France's diplomatic relations. This project could be realized on time only due to the long-term relationship between Dintec and STW. Only because of this relationship, the requirements of the machine manufacturer could be met as technical challenges could be solved through seamless communication.

Authors

Francois Malard
Dintec
www.dintec.eu

Hans Wiedemann
Sensor-Technik Wiedemann
www.sensor-technik.com
info@sensor-technik.de

Tomorrow’s mobile machinery

Control systems usually do not consist of only one ECU anymore – they are composed of a number of different units. The first and obvious idea – connecting all devices to the one and only bus – is often not the ideal solution.

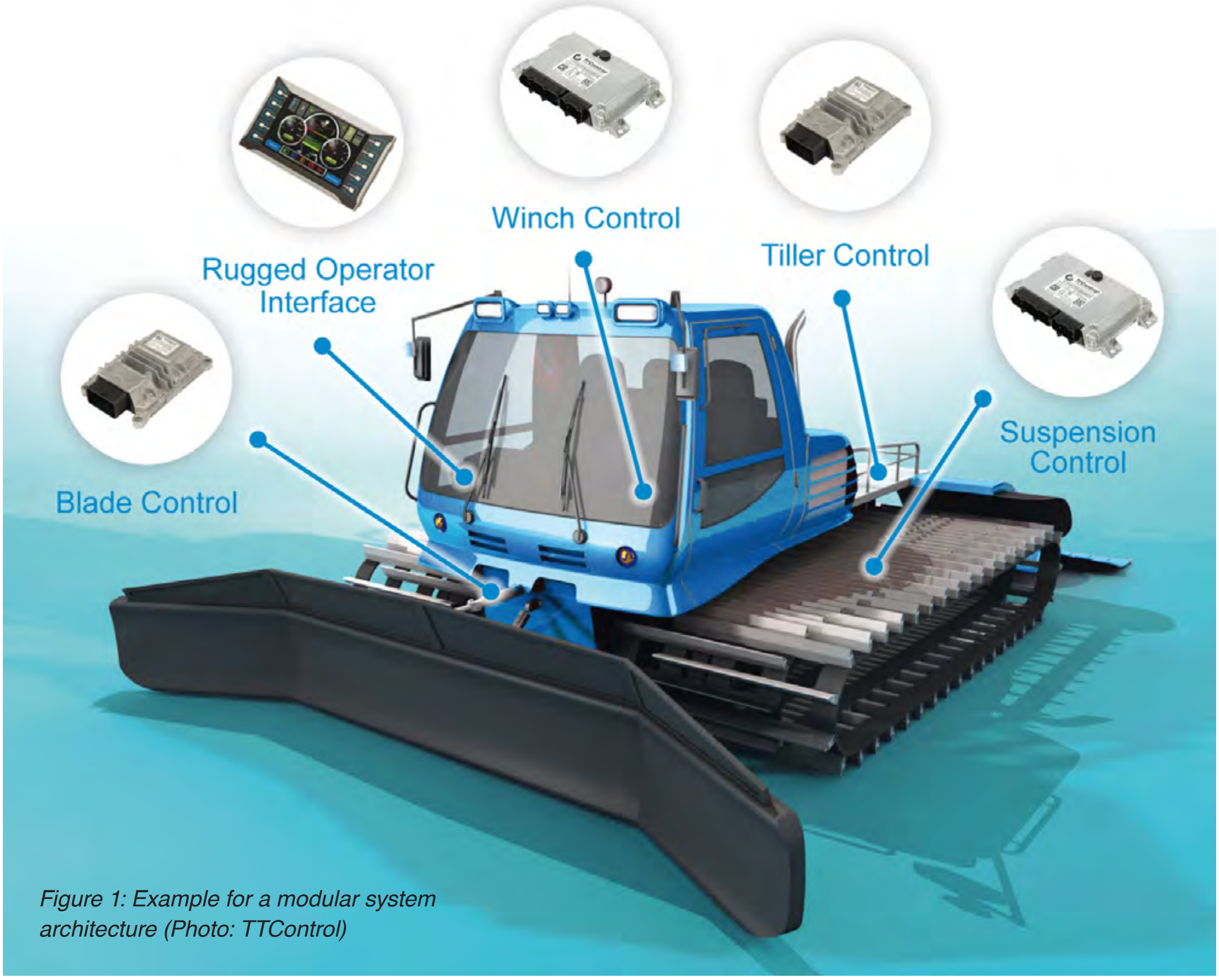


Figure 1: Example for a modular system architecture (Photo: TTControl)

For various reasons, such as reliability, safety, and bandwidth, it might be necessary to:

- connect two ECUs with one (or even more) separate bus(ses) (point-to-point),
- separate “safety relevant” and “comfort features” by different busses,
- offer a debug/monitor interface to the system that is separated from the operational bus.

Aspects like these lead to systems that not only consist of multiple ECUs but also of multiple bus systems for connecting those ECUs. CAN (and its higher-level protocols like CANopen, CANopen Safety, J1939, Isobus…), as a well-established standard for this purpose, is used in uncountable systems today and still is the number one choice control network for mobile machines. This article will not only concentrate on this technology, its strengths and its limitations, but will also point out other technologies that complement the capabilities of CAN.

Why is one ECU not enough?

Many reasons can lead to architectures with more than one ECU. The most common are:

Not enough I/O-pins for the system available on one ECU: The typical solution for this problem is to extend the I/O capability of the master device with as many slave devices (operating as intelligent I/O-modules) as necessary. Even high-end ECUs like TTControl – Hydac International’s HY-TTC 580 with its 96 I/Os – reach their limits in very complex applications. Based on a powerful network concept, this can easily be handled with distributed control systems.

Sensors/actuators spread over long distances: In order to reduce wiring costs or to increase measurement accuracy, it can be helpful to place I/O-modules as close as possible to sensors (this also applies to EMC-sensitive actuators) and connect this unit to the supply and bus lines of the system. This does not only increase the signal quality because of ▷

less noise due to shorter analog connections, it also reduces the wiring effort significantly.

Modularity – usage of existing blocks: Based on already existing modules (such as a multiple-valve block with a CAN-interface), it might be easier and more cost efficient to use a proven-in-use component for a specific purpose and just connect the master control unit via CAN. A generic device optimized for this purpose is the HY-TTC 30 IO-module. If functional safety is required for the I/O slave, the HY-TTC 30XS or HY-TTC 48XS can be used providing CANopen Safety communication.

What has to be taken into consideration when using such an architecture is the fact that for high-dynamic control applications not only a high-performance control unit for executing the control algorithms is needed, but sufficient bandwidth also has to be provided between the master control ECU and multiple I/O-modules. If the CAN bandwidth limitations do not allow for multiple I/O-modules connected to one CAN interface, the master control ECU has to provide multiple CAN interfaces.

Increasing system robustness

For increasing the robustness of a CAN controller network, it is a common practice to separately connect groups of devices to different CAN networks, sometimes even only two devices in a point-to-point-connection to make sure no other device can disturb the communication between the others, for example because of a malfunction.

By following this architecture pattern, it is easy to define several shutdown-levels that – depending on the devices that show a malfunction – can offer at least a „limp home“-mode performed by a couple of high-reliable devices responsible for the core functions of a system. The number of separate CAN networks definitely should not depend on the number of available CAN ports on the master device, but should be based on system architecture and safety aspects. Therefore, a master device with sufficient CAN interfaces is a mandatory precondition.

There are uncountable devices available on the market that can be connected to a CAN network, but still there are some differences, not only when it comes to pure operational functionality, but also concerning commissioning support, variant handling of devices, and enabling easy servicing, to mention just a few of them:

Auto bit-rate detection: With this feature, setting up a network as well as exchanging a device due to maintenance reasons becomes much easier. If, for example, the master device has a preset bit-rate and all other devices then adapt their settings to those defined by the master, it then becomes unnecessary to configure every single participating device upfront to prevent a communication breakdown (bus-off) on the whole network, which happens if there are different participants with different bit-rates in the network. Automatic bit-rate detection also makes it much easier for a debugging/diagnostic interface to connect to a system without any need of communication parameter setup.

Configurable termination: A CAN network needs termination at both ends. Those are usually built into ECUs but if for example three I/O-modules are used (which are 100 %

▷

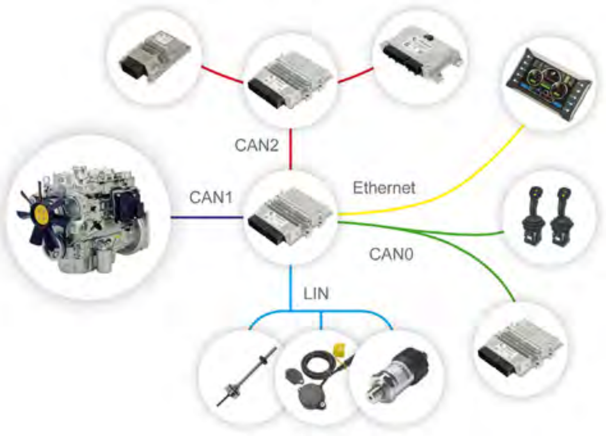


Figure 2: Combining bus technologies – a win-win situation (Photo: TTControl)

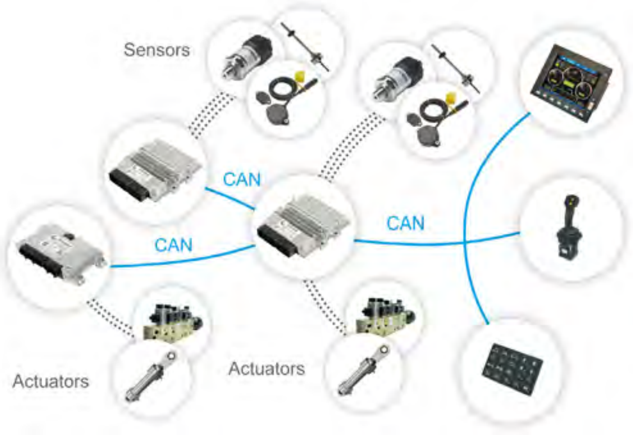


Figure 3: Separate CAN networks for increased robustness (Photo: TTControl)

identical and decide which messages to send and receive based on a location ID that is dependent on the system wiring) but only one of them shall terminate the bus, one can either integrate the termination resistor in the system cabling or the intelligent I/O-module can – again based on the connector wiring – activate or deactivate the integrated termination resistor.

Complementing a CAN network

CAN is a well-established standard for connecting automation devices, but there are other technologies with their individual advantages that can add value to a automation system:

Ethernet: Two aspects make a high-bandwidth connection a good choice in an automation system: First, for debugging, downloading, logging, and a lot of additional supporting activities besides the operational mode of the system, the bandwidth available on an Ethernet connection reduces cycle time for development and maintenance drastically. Second, for the operational mode of a system, higher bandwidth can for example be demanded for visualization, process monitoring or video streaming for rear-view cameras of vehicles.

Combining the advantages of the easy and cost-effective wiring of CAN and the high-speed data communication of Ethernet, automotive physical layer technologies like BroadR-Reach will also soon emerge in off-highway or mobile applications. Demand for deterministic communication technologies can also be fulfilled by Deterministic Ethernet standards such as TSN (Time-Sensitive-Networking) and Time-Triggered-Ethernet (SAE AS6802).

LIN: The network offers a cheap and easy-to-implement bus connection for low-cost-devices with reduced bandwidth demands.

The ideal master control unit therefore not only supports multiple CAN interfaces, but also connectivity for all other communications technologies that are used in the system, at least for the high bandwidth technologies. For example, TTControl – Hydac International's HY-TTC 580 is equipped with seven CAN channels, as well as Ethernet, a serial interface, and LIN.

CAN is nowadays a well-established standard for connecting control units in distributed control systems and will also stay one of the main technologies for connectivity for the near future. There are other communication technologies that can – and in modern systems gradually will – support and complement the CAN interconnectivity between electronic control units for several reasons. One of them is the need for higher bandwidths that also make very powerful ECUs necessary to be capable of handling this large amount of data and also perform gateway tasks in the background while executing controlling applications. ◀

Author

Alexander Schramek
Project Manager Off-Highway
TTControl
www.ttcontrol.com
office@ttcontrol.com

Labview with embedded Linux on ARM

More and more embedded systems develop into distributed, networked systems. To solve this new complexity, Labview combined with embedded Linux and a dual core ARM system-on-module is one useful method.

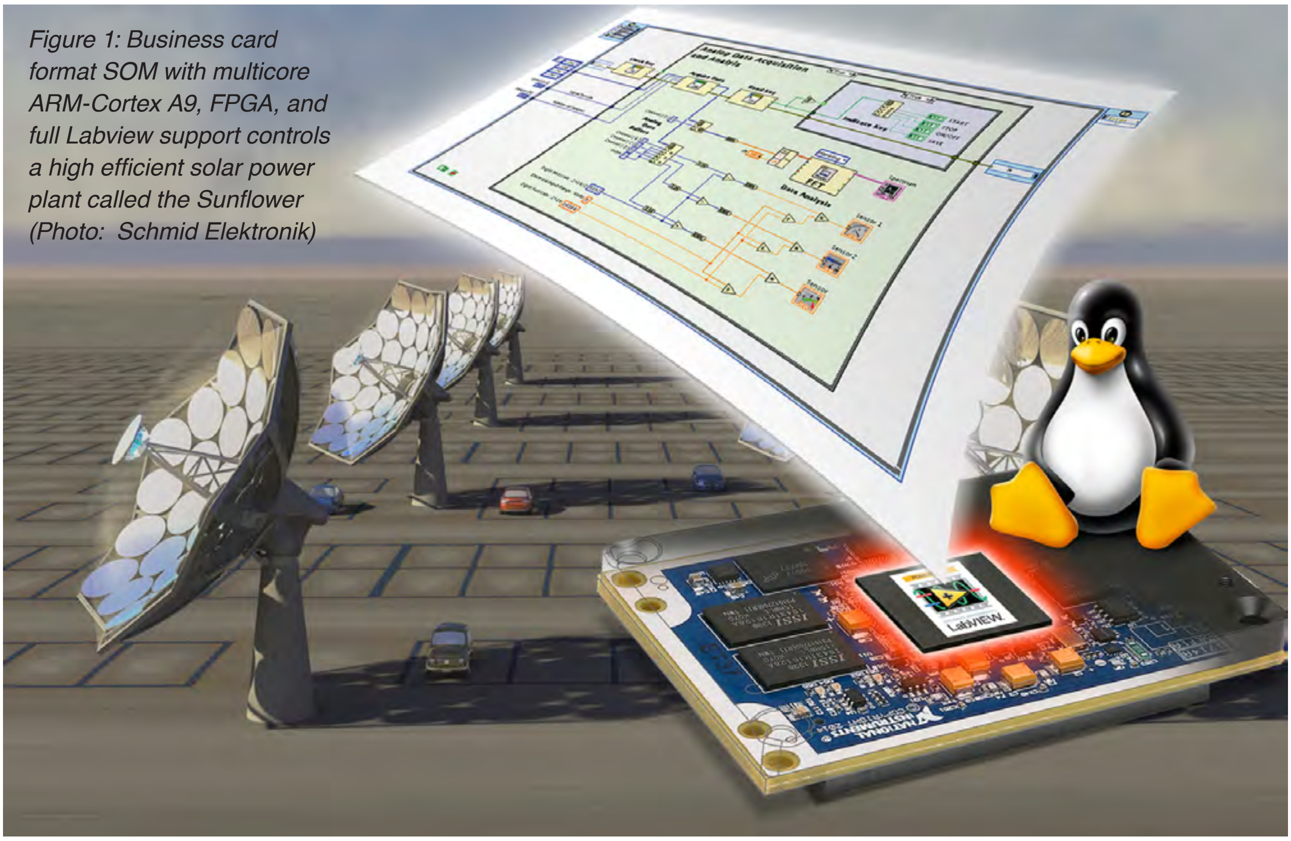


Figure 1: Business card format SOM with multicore ARM-Cortex A9, FPGA, and full Labview support controls a high efficient solar power plant called the Sunflower (Photo: Schmid Elektronik)

Conventional product development with embedded systems usually requires dedicated hardware with specific I/Os. A micro-controller, DSP, FPGA, or a combination of these serves as the brain. The programming languages C and VHDL have been the de facto standard so far, but the industry is changing. The conventional approach is being questioned as more and more embedded systems develop into distributed, networked systems, thus forming a completely new league. New ways of thinking and new techniques are required to manage the related increase in complexity. The high degree of abstraction of the graphical programming language Labview is one efficient approach to this end, combined with the flexibility and stability of Embedded-Linux and the raw power of a dual core ARM system-on-module (SOM) with FPGA.

The RIO SOM by National Instruments is as small as a business card. Its operating temperature range of -40 °C to +85 °C makes it suitable for industrial applications. It is energy efficient and provides the functionalities typically required by smart embedded systems. The high-speed CPU consists of a ZYNQ system-on-chip (SOC) by Xilinx, with a 667 MHz dual core ARM Cortex-A9, Artix-7 FPGA, external 512 MiB DRAM, and 512 MiB flash memory as well as communication functions like Gigabit Ethernet, CAN, USB 2.0 host/device, SD card, and serial interfaces. These are linked to the baseboard via a rugged 320-pin connector, including 160 GPIOs of the FPGA. The baseboard conditions the bare TTL signals and contains further customer specific hardware devices as well as supplies. These features do not differentiate the SOM from conventional embedded systems. However, it can be programmed in Labview, which is a major advantage resulting in a higher development speed for programmers who often work under time pressure.

Development time – the new currency

How can a small team achieve big results? First of all, different programming models can be used as flexibly as a Swiss Army knife, tailored to suit the task or the developer's preference. A control engineering concept, for instance, can be implemented and executed in the Matlab notation. It can be included in the graphical Labview block diagram like differential equations created with model based design. Statecharts on the other hand simplify

▷

the complex processing logic, and a C code interface provides the inclusion of existing C algorithms or direct access to the Linux operating system. Second, Labview programmers directly benefit from support without any further development effort: from libraries for mathematics and signal processing, toolkits (e.g. filters, controllers, sound and vibration, vision, motion) and real-time synchronization of decentralized systems to advanced operating strategies (smartphones, tablets) and data communication (Cloud, real-time Ethernet). Third, the timing, operating system, multitasking, multicore, and underlying hardware are conveniently abstracted. Thus, the Labview application can be downloaded and executed in real-time to own embedded hardware, without any in-depth knowledge of the underlying details.

Flexibility and security of Linux

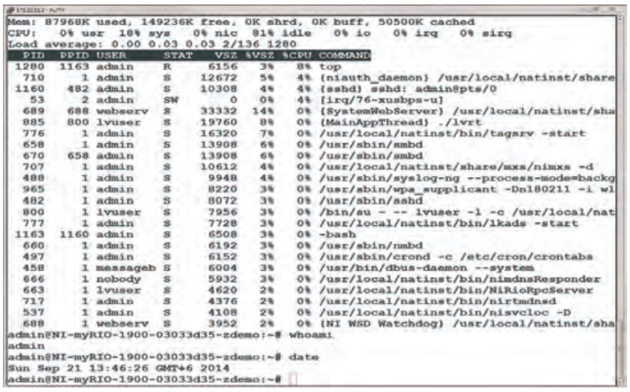
The widely used, popular operating system Linux and the related ecosystem open up new possibilities to the Labview community:

- The Ångström Distribution, optimized for embedded applications, is installed with a repository on the NI servers. The Labview diagram is mapped 1:1 onto the Linux operating system according to the Posix standard. There, the Sandbox of the Linux ecosystem can be deployed, for example, for downloading an SQL database, Apache webserver, or QT GUI with the package manager Opkg.
- The Busybox is a tool for typical embedded tasks, from file system access, system time and, small DHCP client to sleep mode and system reboot.
- Labview has access to the Linux command line, facilitating the direct execution of system commands as well as live management of file system and user permissions.
- Powerful script languages are available, such as Python.
- With TCP/IP, Labview uses Localhost to tap the services of other Linux processes, the so-called daemons.
- The libraries of the Linux operating system can be accessed via the native C API of Labview. In addition, experienced Linux users can configure and recompile the Kernel individually.

U boot – widely used, not only in embedded Linux applications – serves as a bootloader. The kernel is started and initialized based on System V, which facilitates the start of user specific processes. Via one of the shells, the system can be managed by means of an external terminal like Putty (Figure 2). A web-based graphical configuration tool would facilitate the process even further. Instead of using FTP, data is exchanged via WebDAV, an industry standard for secure data transmission on HTTP basis, which is also used by Dropbox.

Particularly in the network of smart embedded systems, timing is the greatest challenge. Timing is an integral element of the language in Labview, as it has always gone hand in hand with industrial process measurement

▷



```
Mem: 87968K used, 149236K free, 0K shrd, 0K buff, 50500K cached
CPU:    0% usr  18% sys   0% nic  81% idle   0% io   0% irq   0% sirq
Load average: 0.00 0.03 0.03 2/136 1280
  PID  PPID USER     STAT    VSZ %VSZ %CPU COMMAND
 1280  1163 admin    R      6156   3%   8% top
  710     1 admin    S     12672   5%   4% {niauth_daemon} /usr/local/natinst/share
 1160   482 admin    S     10308   4%   4% {sshd} sshd: admin@pts/0
   53     2 admin    SW        0   0%   4% [irq/76-xusbps-u]
  689   688 webserv  S     33332  14%   0% {SystemWebServer} /usr/local/natinst/sha
  885   800 lvuser   S     19760   8%   0% {MainAppThread} ./lvrt
  776     1 admin    S     16320   7%   0% /usr/local/natinst/bin/tagsrv -start
  658     1 admin    S     13908   6%   0% /usr/sbin/smbd
  670   658 admin    S     13908   6%   0% /usr/sbin/smbd
  707     1 admin    S     10612   4%   0% /usr/local/natinst/share/mxs/nimxs -d
  488     1 admin    S      9948   4%   0% /usr/sbin/syslog-ng --process-mode=backg
  965     1 admin    S      8220   3%   0% /usr/sbin/wpa_supplicant -Dnl80211 -i wl
  482     1 admin    S      8072   3%   0% /usr/sbin/sshd
  800     1 lvuser   S      7956   3%   0% /bin/su - -- lvuser -l -c /usr/local/nat
  777     1 admin    S      7728   3%   0% /usr/local/natinst/bin/lkads -start
 1163  1160 admin    S      6508   3%   0% -bash
  660     1 admin    S      6192   3%   0% /usr/sbin/nmbd
  497     1 admin    S      6152   3%   0% /usr/sbin/crond -c /etc/cron/crontabs
  458     1 messageb S      6004   3%   0% /usr/bin/dbus-daemon --system
  666     1 nobody   S      5932   3%   0% /usr/local/natinst/bin/nimdnsResponder
  663     1 lvuser   S      4620   2%   0% /usr/local/natinst/bin/NiRioRpcServer
  717     1 admin    S      4376   2%   0% /usr/local/natinst/bin/nirtmdnsd
  537     1 admin    S      4108   2%   0% /usr/local/natinst/bin/nisvcloc -D
  688     1 webserv  S      3952   2%   0% {NI WSD Watchdog} /usr/local/natinst/sha
admin@NI-myRIO-1900-03033d35-zdemo:~# whoami
admin
admin@NI-myRIO-1900-03033d35-zdemo:~# date
Sun Sep 21 13:46:26 GMT+6 2014
admin@NI-myRIO-1900-03033d35-zdemo:~# 
```

Figure 2: The SSH (secure shell) facilitates the management of Linux on ARM and the use of various command line services (Photo: Schmid Elektronik)

and control hardware. Graphical multitasking programming requires operating system functionalites like scheduling. Six schemes are available to this end. Cron facilitates the execution of repetitive tasks down to [min] resolution, e.g. deletion of logfiles or frequent e-mail checks. The CFS (completely fair scheduler) mainly serves to implement work tasks that are not time-critical but still efficient. If [ms] response times are required, the kernel can be configured as preemptive. For time-critical tasks with a required jitter of 10 *µ*s to 100 *µ*s, the Linux kernel is patched with "PREEMPT_RT". The FPGA is used for hard real-time in the one-digit [*µ*s] or even [ns] range. Due to Labview's multicore support, graphical tasks can be directly assigned to a processor core.

Programming the FPGA with a mouse

The Artix-7 FPGA hardware is software-reconfigurable and suitable for parallel processing (Figure 3). Time-critical tasks and I/Os are delegated to this hardware to reduce ARM processor load. Practicable functionality as well as the related timing are two crucial parameters. So far, the programming of FPGAs has required specific expertise. With Labview, even engineers without such expertise get access to the powerful technology of reconfigurable logic. Using intuitive functional blocks, a range of analog, digital, and serial process signals can be integrated, linked, and preprocessed in parallel before being transferred to the ARM processor. The functionality of the application to be mapped is directly limited by the number of available gates. Consequently, available information on how many gates are required for each specific operation (addition, filter, FFT) is a valuable basis of decision-making regarding the functional scope that can be implemented as a maximum. In the context of timing, even application programmers with graphical programming preference think in ticks, the smallest FPGA time increment in the nanosecond range. It defines how much time is required by logical and mathematical operations and I/O accesses, providing target values for system timing.

In the software design process, the embedded application has to be split between ARM and FPGA. The ARM processor is in charge of high-level main functions. Low-level details like device drivers, time-critical code, digital filters, combinational and sequential logic, scaling, fixed point, and integer arithmetic, are preferably outsourced to the FPGA. The NI Labview FPGA diagram is synthesized in VHDL code, compiled to firmware/bitfile by the FPGA tools and downloaded to the FPGA.

Almost any commercially available I/O component can be connected to the SOM through a baseboard (Figure 6) and controlled with Labview, e.g. via digital I/Os, synchronous (SPI, I²C) and asynchronous (UART) serial interfaces or parallel high-speed bus systems. Typical examples are analog I/O, PWM, counters, encoders and digital I/O, wireless/WLAN, RFID, GSM/GPRS, GPS, Zigbee, and color TFTs with CAP/multi-touch. The hardware can be adapted to any task as regards shape and function. To achieve such flexibility, the first step is to develop hardware in the form of a baseboard, a task usually required in a dual-plate approach. The most critical circuits around the CPU and memory are already implemented on the plugin SOM.

Individual hardware drivers with Eclipse

In the Labview application, external I/O devices are activated through intuitive virtual instruments (VIs). Low-level drivers are required to this end. If the devices are connected to the 160 FPGA pins, drivers can be implemented with a similar technique as for Compact RIO. If they are connected directly to the ARM processor and the BSP (board support package) has no driver, Linux offers the following three options:

- If the driver is available as executable, it can be executed in Labview directly through the command line VI. Here, Labview is logged in as "lvuser".
- Otherwise, the executable is generated with the Eclipse IDE (Figure 4) and executed like in the first option. To this end, the driver has to be available as C/C++ source code. Due to access through the operating system, response times in the two-digit millisecond range can be achieved.
- A library (shared object) is generated with Eclipse and addressed in Labview via the C API (Figure 5), similar to a DLL under Windows. Compared to the both previous options, direct access to the C library now facilitates timings in the two-digit µs-range.

System-on-module in a solar power plant

With the Sunflower, a Swiss company brings solar energy to the remotest places on earth. No matter where you live,

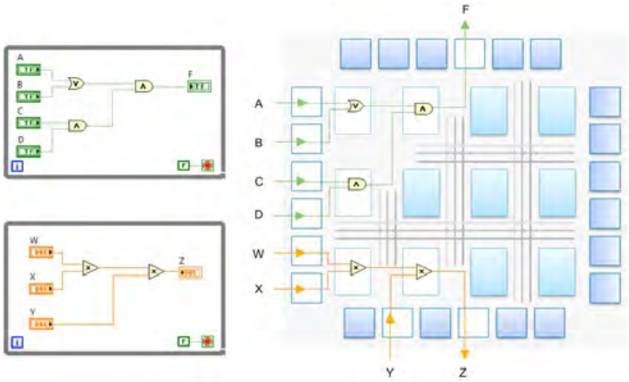
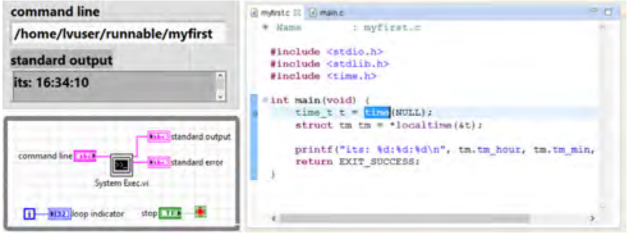


Figure 3: The true parallelism of FPGA programming comes closest to the Labview data flow paradigm (Photo: National Instruments)



command line
/home/lvuser/runnable/myfirst
standard output
its: 16:34:10

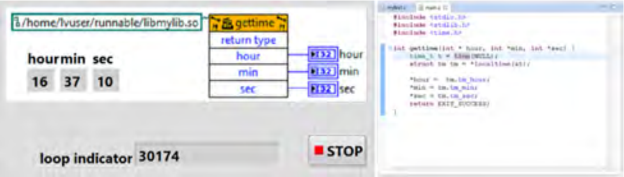
```
 * Name        : myfirst.c

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void) {
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);

    printf("its: %d:%d:%d\n", tm.tm_hour, tm.tm_min,
    return EXIT_SUCCESS;
}
```

Figure 4: Labview executes a Linux executable compiled in Eclipse (right) via the command line VI (virtual instrument, left) (Photo: Schmid Elektronik)



/home/lvuser/runnable/libmylib.so

hour	min	sec
16	37	10

loop indicator 30174

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int gettime(int * hour, int *min, int *sec) {
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);

    *hour =  tm.tm_hour;
    *min = tm.tm_min;
    *sec = tm.tm_sec;
    return EXIT_SUCCESS;
}
```

Figure 5: Through its C API (application programming interface, left), Labview accesses a Linux library generated in Eclipse (.so = shared object, right) (Photo: Schmid Elektronik)*

you get cooling in summer, heating in winter, clean drinking water, and even fuel. The key of the solar plant is to concentrate solar radiation in the focal point of a dish with a factor of 2000 and extract energy with an efficiency of 80 %. This is achieved with ultra-high efficiency solar cells and a cooling system of IBM. The Sunflower weighs 18 t, is 10 m high and produces up to 300 kWh of energy during a sunny day. It consists of three main elements: the optics, the receiver, and the tracker.

The optics is represented by a huge 40-m^2 dish. Its main function is to concentrate solar rays in its focal point. To achieve this, the dish contains 36 elliptic mirrors. On each of these mirrors, a very thin, silver coated foil similar to chocolate wrappers is applied through a very small vacuum. That's why they are also called pneumatic mirrors. A solution like this is much lower in cost compared to warped solid state mirrors. The dish is covered with an inflated, robust plastic membrane. The reason for this is twofold: First, the dish has to be protected from the environment such as rain, sand, and other dirt in order to keep the mirrors clean. Second, to protect wildlife such as birds from the sunflower. The inflation is achieved with a pneumatic system consisting of fan blower, compressor, and an accumulator that are located in the pole of the sunflower.

An sbRIO-9651 SOM as the system's brain moves the dish continuously towards the sun and connects to dozens

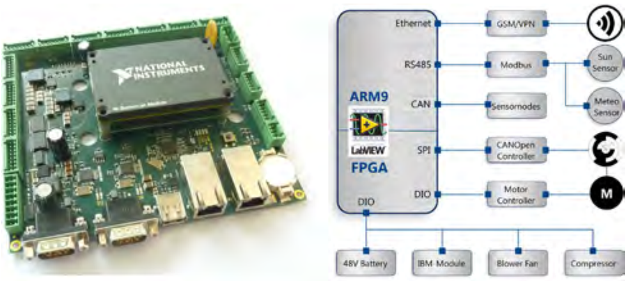


Figure 6: The SOM (black) is plugged onto a baseboard (left) of an individual form factor, which contains customer specific hardware (right) (Photo: Schmid Elektronik)

▷

of sensors through a baseboard (Figure 6). Stepper motors move the two axes of the tracking system using absolute encoders in the feedback loop. In order to keep production costs low, the motor driver has been implemented with discrete electronics such as transistors to drive the PWM. The tricky thing to handle was to connect to the encoder using CANopen to the FPGA. The solution is a micro-controller, featuring CANopen in its silicon, connecting to the system-on-module with SPI and acting like a translator. On top of that, the CAN network of the ZYNQs ARM, linked with Labview Realtime, was also connected to the CANopen network. This link is mainly for servicing the CANopen sensors during configuration mode. As for the encoder, this meant configuring the sampling rate of the encoder signals on the CAN network.

Next, a commercially available sun sensor is connected via Modbus RTU. A local weather station delivers the local temperature, humidity, pressure, wind force, and direction over the TCP/IP-Modbus. The VPN access is realized with a netmodule, connected via Gigabit Ethernet. A distributed sensor network including pressure-, temperature-, and humidity sensors is connected by CAN and helps the controlling unit to maintain a stable climate within the covering membrane. All other devices such as the fan blower, compressor, and electronic valves are controlled by a robust 24-V PLC type of digital signals. ◀

Author

Marco Schmid
Schmid Elektronik
www.schmid-elektronik.ch
info@schmid-elektronik.ch

80 % energy efficiency from solar concentration

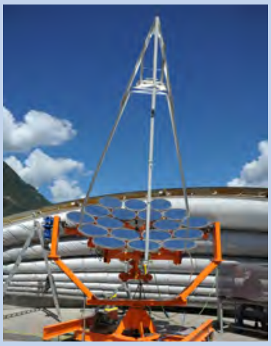
Airlight Energy, a Swiss-based supplier of solar power technology, has partnered with IBM Research to bring affordable solar technology to the market. Their High Concentration Photovoltaic Thermal (HCPVT) system can concentrate the sun's radiation 2000 times and convert 80 % of it into useful energy to generates 12 kW of electrical power and 20 kW of heat on a sunny day — enough to power several average homes. The stand-alone dish provides electricity, heat, hot water, and conditioned air (through a chiller) all at the same time. The system, which resembles a sunflower, uses a parabolic dish made of patented fiber-based concrete. The concrete can be molded into nearly any shape in less than four hours and has mechanical characteristics similar to those of aluminum at one-fifth the cost. The photovoltaic chips, similar to those used in orbiting satellites, are mounted on micro-structured layers that pipe treated water within fractions of millimeters of the chip to absorb the heat and draw it away 10 times more effectively than with passive air cooling. The +85 °C to +90 °C hot water maintains the chips at safe operating temperatures of +105 °C, which otherwise would reach over +1500 °C. The entire system sits on a sun tracking system, which positions the dish at the best angle throughout the day to capture the sun's rays.

"The direct cooling technology with very small pumping power used to cool the photovoltaic chips with water is inspired by the hierarchical branched blood supply system of the human body," said Dr. Bruno Michel, Manager of advanced thermal packaging at IBM Research. With such a high concentration and based on its radical design, researchers believe that with high-volume production they can achieve a cost two to three times lower than comparable systems. Based on its current design, scientists estimate that the operating lifetime for the HCPVT structure will be up to 60 years with proper maintenance. The protective foil and the plastic elliptic mirrors will need to be replaced every 10 to 15 years depending on the environment, and the photovoltaic cells need replacing every 25 years.

Greenpeace named the Sunflower the number one solar wonder of the world because it can "not only provide electricity – it can also desalinate water for sanitation and drinking. A group of several solar generators could provide enough fresh water for an entire town. The sunflower operates by tracking the sun, so that it always points in the best direction for collecting the rays – just like a real sunflower!" The system can be customized with further equipment to provide drinkable water and air conditioning from its hot water output. For example, salt water can pass through a porous membrane distillation system, where it is vaporized and desalinated. Such a system could provide 30 l to 40 l of drinkable water per square meter of receiver area per day, while still generating electricity with a more than 25 % yield or 2 kWh per day – a little less than half the amount of water the average person needs per day according to the United Nations, whereas a large multi-dish installation could provide enough water for a town.

Scientists at Airlight and IBM envision the HCPVT system providing sustainable energy to locations that need both outputs of the sunflower: energy and heat. Possible locations around the world include southern Europe, Africa, the Arabian peninsula, the southwestern part of North America, South America, Japan, and Australia. In addition to residences, additional applications could include remote hospitals, medical facilities, hotels and resorts, and shopping centers.

Annegret Emerich



Youtube
Airlight Energy and IBM Bring Solar Electricity and Heat to Remote Locations

Customized CANopen tools made easy

Instead of inventing the wheel again and again, standardized, CANopen compliant API may be used independently of programming environments and applications.

There are many kinds of CANopen equipped applications on the market. Most of them are test and commissioning tools for individual devices and entire systems. Traditional implementation approach has been either to implement separately only some CANopen services or link an entire CANopen stack into an application as a source code. The first approach has been typically frame-oriented, and thus lead into very constrained functionality and regular tool updates, according to the updates in the corresponding device. Latter approach typically has not to be updated so often, but the use of a CANopen stack as a source code with all possible extensions in the application-programming interface (API) may lead into unnecessary complexity. Both alternatives do not help in the development of next products, because of the constrained or complicated API usage. Moreover, application specific implementation leads into testing the same basic CANopen features from application to another.

However, sufficient CANopen API for any kind of purposes can be kept very simple. Only the standardized communication services may be considered. In commissioning tools service data objects (SDO) play a primary role, optionally in conjunction by the heartbeat consumer. Commissioning tools may additionally require process data object (PDO) support, because fine tuning of some runtime parameters often need to be adjusted. Furthermore, an emergency consumer is needed in order to provide comprehensive state management in conjunction with the other services.

This article presents an approach with CANopen implemented as highly re-usable library external to the application. Python is used as an example application programming language. A concept, how the external CANopen library concept works is presented after details behind selection of Python and before the details. The first detailed section presents main differences between IEC 61131-3 and Python. The second detailed section introduces object access mechanisms. Next detailed sections describe, how signals differ from parameters and how signals and parameters may be accessed consistently. The last section sets concluding remarks and proposes some further development.

Python

Python was chosen as an application development environment, because of its numerous advantages. It is available for free and there exists a large community providing support and knowledge pool in various discussion forums. Furthermore, it is a very productive high level language. Due to the large community, the large number of free libraries for various purposes, increase the development efficiency further. The operating system (OS) independence provides not only support for several operating systems, but also an easier upgrade path between operating system versions, which has been a major problem especially between various Microsoft Windows versions.

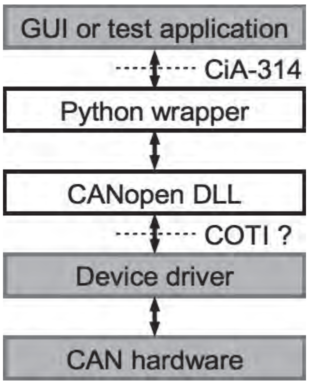


Figure 1: Software layers

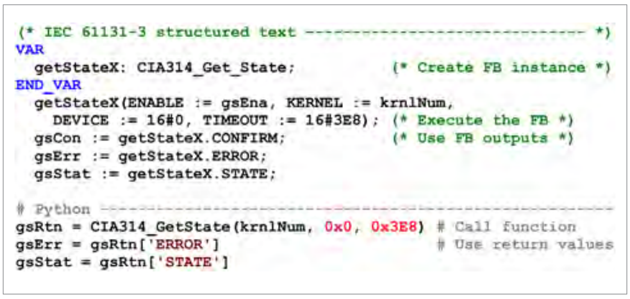
Python also supports OS-independent GUI implementations. There are various graphical user interface (GUI) toolkits available. Tcl/Tk based Tkinter is built-in into the Python environment. There are also additional widgets available for Tkinter, which are supported by Tcl/Tk, but not by Tkinter by default. There are also QT-based toolkits for those who prefer QT as a GUI environment. When OS-independence is important, special attention shall be paid on finding out the potential issues with QT among the operating system.

CANopen library concept

The entire CANopen implementation has been encapsulated into an independent library with a standardized API. It was found, that API according to CiA 302-4 [5] and CiA 314 [6] works extremely well, independent of the programming language. The main advantage of the library implementation is, that it works like a CANopen interface of a programmable logic controller (PLC), which enables a direct use by PLC programming experts, without learning lots of new things. Newcomers need to learn only a single API, which applies for both supporting tools and PLCs. Another significant advantage of such library is, that due to a re-use of it with various applications, it will be continuously tested. It has been proved, that the use with various applications rapidly reveals the problems, due to the large number of use cases.

Everything with the external library works in a CANopen way – through the object dictionary. In addition, due to the standardized API [6], everything may be adjusted also from the application – through the object dictionary. In the case of various testing and commissioning tools, there do not exist fixed configurations and applications that shall scan the system and adjust available services accordingly.

▷



```
(* IEC 61131-3 structured text ---------------------------- *)
VAR
  getStateX: CIA314_Get_State;          (* Create FB instance *)
END_VAR
  getStateX(ENABLE := gsEna, KERNEL := krnlNum,
    DEVICE := 16#0, TIMEOUT := 16#3E8); (* Execute the FB *)
  gsCon := getStateX.CONFIRM;           (* Use FB outputs *)
  gsErr := getStateX.ERROR;
  gsStat := getStateX.STATE;

# Python ------------------------------------------------------
gsRtn = CIA314_GetState(krnlNum, 0x0, 0x3E8) # Call function
gsErr = gsRtn['ERROR']                       # Use return values
gsStat = gsRtn['STATE']
```

Figure 2: Example call of IEC 61131-3 FB (top) and corresponding Python function (bottom)

On the contrary, the utilization of fixed configurations apply for the GUI applications.

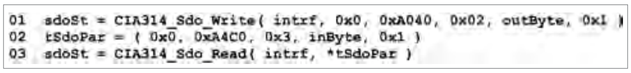
Signal usage complies with PLCs, where application contains global variables for storing the temporary values over a single application cycle. Output variables are written to the process image [5] at the end of each cycle and input variables read from it at the begin of each cycle. An additional Python wrapper module has been implemented only in order to keep applications as simple as possible by adapting C-style data types, used in the library, into Python data types in a dedicated module.

IEC 61131-3 vs. Python

Unlike IEC 61131-3, Python does not support function blocks (FB), that is, why functions shall be used instead. The main difference between FBs and functions is, that execution control signals Enable and Confirm are not needed in functions. All other arguments and return values have been kept intact. It has been considered, that the function based API could be used also in other programming languages with minimal changes.

Function arguments are passed in the simplest way, fixed number of arguments in a fixed order. Arguments for Python functions may alternatively be passed as named values, which would allow passing of unsupported arguments without leading into exception. This would also allow passing of subset of arguments without exception, which is not supported in IEC 61131-3. Named arguments are related with the dictionary data type, which is not supported by other common languages. Thus, fixed order and fixed number of arguments are used. When the arguments are provided as a tuple, they can be passed to the appropriate function as a single parameter. Figure 2 clarifies the minor differences.

IEC 61131-3 function blocks may return multiple values, which is not directly supported by functions of more traditional programming languages. Python supports the dictionary data type, which is excellent for the return values – multiple return values can be provided as dictionary members and are accessible by key names, which is almost equal to the IEC 61131-3 function blocks from access point of view. Same principles are used throughout the Python implementation of the API as specified in CiA 314. Data structures may be used accordingly with other languages.



```
sdoSt = CIA314_Sdo_Write( intrf, 0x0, 0xA040, 0x02, outByte, 0x1 )
tSdoPar = ( 0x0, 0xA4C0, 0x3, inByte, 0x1 )
sdoSt = CIA314_Sdo_Read( intrf, *tSdoPar )
```

Figure 3: Example SDO operations with separate arguments and arguments packed into a tuple

▷

A further possibility in Python is, that multiple arguments may be packed into a tuple and be passed to a function as a single group of arguments. Such a mechanism enables the collecting of object access information into a single location, where it is easy to maintain. Majority of the application code remains simple and independent of the system configuration. Example of this exists in Figure 3.

Object access

One of the main tricks with the CANopen library is, that also objects in the local object dictionary are accessed with SDO functions. The primary result is, that such a mechanism complies with accessing any kind of functions from an external library, which conforms the way of working in most operating systems. Direct memory access, which is traditionally used as signal object accesses in PLCs, is not a preferred mechanism in major operating systems. Thus, additional interfaces are not required and a well-supported object dictionary (OD) based interface concept may be used.

A write operation of byte value of outByte into local network variable 0xA040:0x02 with individual arguments is shown in line 1 of Figure 3. A tuple defining read arguments of byte value of local input network variable 0xA4C0:0x03 is composed in line 2. Actual read operation is shown in line 3, after which the inByte variable contains the read value. Type of both outByte and inByte shall be defined according to the object type, c_ubyte in the example.

Especially in applications with GUI or log file, it is useful to use SDO abort code decoding function for converting numeric codes into text descriptions. Standardized abort codes are available in machine understandable format, already translated into several languages. It has been presented, how this is read and translated into a decoding function [3]. The optimal implementation varies, depending on the capabilities of the used programming language.

Signal and parameter objects

From an application point of view, most significant difference between signals and parameters is, that parameters are accessed only on-demand but signals shall be regularly updated between application variables and process image in the OD. Periodically called functions read input values from the library and write output values to the library. Such an approach results a minimum number of updates and maximum performance, when only values of the used objects are read and written. If the application accesses different set of signals in different modes, unused signals need not to be updated.

Related articles

- Optional structures in CANopen projects
- CANopen device configuration editors
- SI-Unit and scaling management in CANopen
- Exception management in CANopen systems

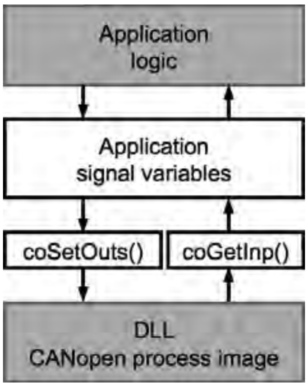
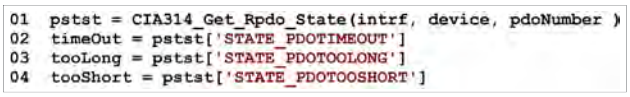


Figure 4: Process image synchronization with periodically called functions coSetOuts() and coGetInp()

Parameter objects are typically managed in an application dependent manner, sometimes individually or more often in groups but almost never all at a time, especially in the larger systems. Parameter accesses shall be designed carefully so, that they do not have significant impact on the network schedule [1]. Thus, a general approach for exporting parameter read-and-write functions cannot be found.

Signal consistency

The typical heartbeat transmission interval is several hundreds of milliseconds and thus it may either be executed in an own thread or sequentially read as part of signal synchronization thread one node-ID in each cycle. One should remember, that reading NMT states shall not be dependent on the NMT-state, because problems in network structure are most important use cases for system monitoring. Heartbeat provides an raw operational state of each device. Such information gives indirect validity indication for incoming signals – if the source device is not in operational state or even does not exist, corresponding RPDOs cannot be valid in any case.



```
pstst = CIA314_Get_Rpdo_State(intrf, device, pdoNumber )
timeOut = pstst['STATE_PDOTIMEOUT']
tooLong = pstst['STATE_PDOTOOLONG']
tooShort = pstst['STATE_PDOTOOSHORT']
```

Figure 5: Example of getting RPDO status

Status of incoming signals is essential in control systems and importance increases among the increasing safety requirements [2]. RPDO status is a standardized basic service [4] and its results are as easy to be distributed to the mapped signals equally to the heartbeat status from producer devices to the corresponding signals [1]. A standard API does not exist yet, but a call shown in Figure 5 is under development and will be proposed into CiA-314.

Additional cross-reference is required for distributing the PDO status into meta-information of the mapped signal. Such information is already available in the RPDO mapping objects of standard DCF files, when a network project has been completed. Distribution may be located into a dedicated abstraction layer service function with heartbeat consumer information [1].

Parameter consistency

Parameter handling presented in the literature [1] applies also to the Python approach without any changes. A special ▷

References

[1] Saha H., Improving development efficiency and quality of distributed IEC 61131-3 applications with CANopen system design, Proceedings of the 13th iCC, CiA, 2012

[2] Hietikko M., Malm T., Saha H., Comparing performance level estimation of safety functions in three distributed structures, Journal of Reliability Engineering and System Safety, issue 134, Elsevier, 2015, pp. 218-229

[3] Saha H., CANopen in the frontline of open data, CAN-Newsletter 3/2014, CiA, 2014, pp. 42- 45

[4] CANopen application layer and communication profile, CiA-301, CiA

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

[6] Accessing CANopen services in devices programmable in IEC 61131-3 languages, CiA-314, CiA

[7] COTI, Conformance Test Interface, Appendix to CANopen Device test, CiA, 2008

[8] LIN Specification Package, Revision 2.2A, LIN Consortium, 2010

case is, that there are two or more devices at the same node-ID in the network. Heartbeat reveals unambiguously only missing nodes, not necessarily case, when two or more devices share the same node-ID. Detection of duplicate devices is typical in various commissioning tools, but may also be required in control systems.

In the case of two nodes with the same node-ID, the SDO client receives the first server response normally, but raises an exception if the second SDO server reply is received without a client request. Such an error condition shall be read by function CIA314_Get_CANopen_Kernel_State(), because in such a case the SDO transaction seems to end normally and CIA314_Sdo_Read() or CIA314_Sdo_Write() does not return any error status.

Depending on the implementation, the status may also need to be called before the SDO function, in order to clear the potential error status caused by preceded operations. Meta information earlier presented in the literature [1] have been supplemented by the status attribute, where the SDO status will stored.

Conclusions

Independent CANopen library with Python and Tkinter provides very efficient GUI implementations, mostly thanks to the Tkinter layout manager taking care of many basic widget operations. The CANopen library provides similar easy-to-use approach by implementing system integration interface similar to a PLC with CiA-302-4 compliant process image and CiA-314 compliant CANopen service access functions.

CANopen API specified by CiA-302-4 and CiA-314 is tiny, but still sufficient for interfacing the CANopen system. The use of harmonized API among PLCs and GUIs causes less to learn by programmers. Concise API results in fewer opportunities to make mistakes and less code to be maintained and tested. During two years of development work with various applications, more complex API have not be required.

Cyclic signal update according to the GUI update rate helps in avoiding overloading GUI framework by network reception events, which is a common result with callbacks. The cyclically updated variables approach introduces only a load required by current application state.

While a standardized API exists for CANopen, re-usability of the approach could be improved further by developing a generic hardware abstraction layer (HAL) based on the CANopen conformance test interface (COTI) specification [7]. Such an approach has been used with Local Interconnect Network (LIN) from the very beginning [8]. ◀

Author

Heikki Saha
TK Engineering
www.tke.fi

EMC effects underestimated as fault causes

In many machinery and equipment, CAN is the backbone of communication technology. Despite this, bus systems are often not given the attention they deserve in preventive maintenance.

At the same time topics as Industry 4.0 and the Internet of Things (IoT) are bringing to the fore topics which result in an increasing degree of cross- linking. To avoid the risk of failure, you need to act now. If we take Industry 4.0 to the next logical step, the whole production will be order-related for the customer. Problems will disrupt not just the impacted machine, but the entire chain. Important intermediate storage facilities, which can supply products in the period of interruption are no longer foreseen. The increasing degree of cross-linking is also increasing power density and therefore susceptibility – for example by electromagnetic influences. This emphasizes the importance of a stable field bus communication.

The umbrella term used is Electromagnetic Compatibility (EMC). EMC considers whether electrical devices and networks themselves disturb other components (interference source) or are disturbed by other components (interference sink). The goal is therefore to construct all electrical equipment so that it doesn't disturb others and cannot itself be disturbed.

Types of disturbance

The electromagnetic influence between the interference source and interference sink is known as coupling. A distinction is made between:

- Direct coupling - Conductive connection between two circuits, usually by means of shared supply or ground line
- Capacitive (electrostatic) coupling - Mutual influence by the electrical field, for example, by conductors located close to each other with a high potential difference
- Inductive (magnetic) coupling - The alternating field generated by a conductor's current flow induces a disturbance voltage in other circuits
- External disturbance source - For example, lightning strike, should be noted in particular if the cables in extensive installations are routed outdoors

Disturbed serial bus systems

While initially a telegram bit fails occasionally, as the situation worsens, regular destruction of complete telegrams may occur. The bus communication failure is often caused by damage on the bus installation along with electromagnetic effects. These have a major influence on the data traffic and during operation result in gradual curtailments, culminating in a plant standstill. "When we are called to equipment stoppages, in over half of all cases, we find EMC problems," tells Hans-Ludwig Göhringer from IVG Göhringer. For many years, the company has maintained serial bus systems, such as CAN, and is now a recognized expert in this field. IVG Göhringer is often called in to troubleshoot equipment stoppages and shares the experiences it has gained in various training courses.

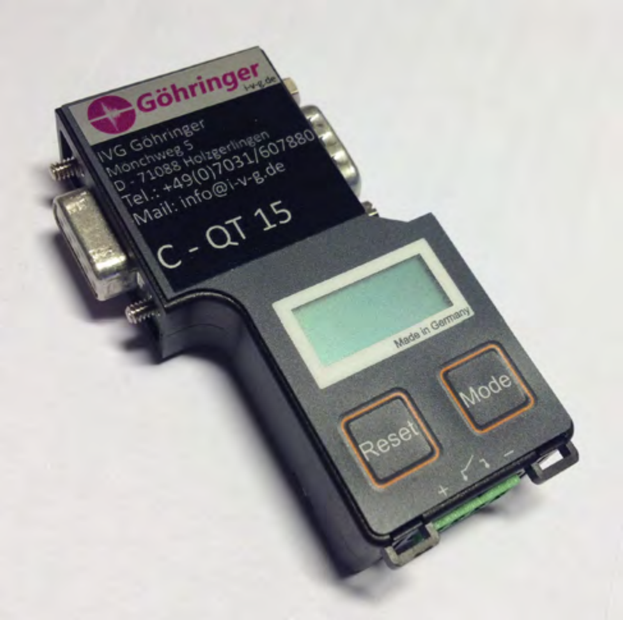


Figure 1: Errors in CAN communication are indicated by the integrated status LEDs and a potential-free alarm contact (Photo: IVG Göhringer)

Finding causes

When trying to find the reason what has caused faults, it should be distinguished between design shortcomings and bus installation ageing. However, compared with the situation ten years ago, we are currently seeing much more emphasis being placed on equipment design which takes EMC into consideration.

Design deficiencies include:

- Inferior quality plug connectors rather than industrial quality
- Shielding only being fitted on one side
- Pigtail shielding rather than connections covering the entire surface
- No potential compensation or potential compensation of an inadequate size
- Power and data cables not kept separate
- Neutral earthing rather than meshing

▷

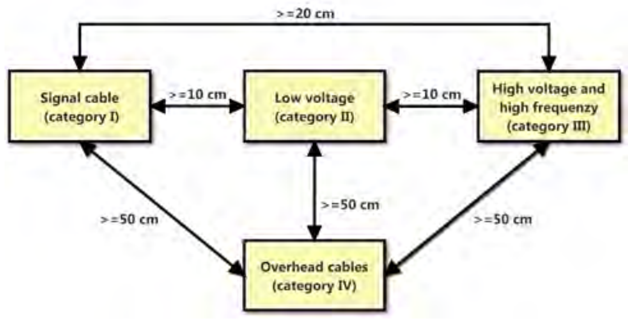

>=20 cm
Signal cable (category I)
>=10 cm
Low voltage (category II)
>=10 cm
High voltage and high frequenzy (category III)
>=50 cm
>=50 cm
>=50 cm
Overhead cables (category IV)

Figure 2: Minimum spacing and distances between different categories of cable (Photo: IVG Göhringer)

“When looking for the components responsible for faults, we first consider switching contactors and inverters with high outputs and correspondingly high currents,” said Hans-Ludwig Göhringer, adding: “But there are many other components which are needed for functional processes and may also be the cause of the problem.” These include:

- frequency converters
- motors and brakes
- photovoltaic systems
- coils
- fluorescent lamps
- heaters
- switching power supplies, converters
- switches, contactors
- wireless sections
- magnetic alternating fields
- static discharges, arcs

The requirements of electric automation are also on the rise. Faster speeds in the equipment require shorter switching times and greater control accuracy when positioning. “We are seeing more and more switching sequences and steeper flanks, meaning that high-frequency faults are increasing too,” explains Hans-Ludwig Göhringer.

Shielding is important

The most important way of protecting machinery and equipment from electromagnetic faults is proper shielded cables and connections. This includes a shield connection covering the entire surface and earthing at both ends. Every now and then we see shielding and shielded cables that are only soldered on at one point. So the shielding is ineffective, especially at high frequencies. The shielding is only fit for purpose if it is continuous, closed from one end to the other and is also connected to the functional earth with good conductivity. The use of metallic cable bushings prevents high-frequency faults from penetrating controllers and switch cabinets. “Sometimes the shielding is only used on one side as it is argued that no current can flow on the shielding,” reports Hans-Ludwig Göhringer, adding: “But that is nonsense. A high shield current implies there is no potential equalization – that is where you have to start.” Which brings us to the next issue.

EN 50310 sets out the minimum requirements for earthing and potential equalization for buildings with IT facilities, including electric control technology, bus systems and networks. We would always recommend changing from neutral earthing to a meshed earthing system.

▷

Figure 3: Hans-Ludwig Göhringer passes on his knowledge of maintaining bus systems and networks in workshops and training sessions (Photo: IVG Göhringer)

This standard may have been produced in the context of Ethernet cabling, but it is useful for all other bus systems too. "The main idea behind the meshed structure is that the current finds the right route," explains Hans-Ludwig Göhringer, adding: "In principle, this route is the right one. But there is no single solution that is suitable for all equipment. Even with textbook meshed earthing, instances may arise where the current gets somewhere you didn't want it." Furthermore, the corresponding cable cross-sections aren't defined in EN 50310. A structured approach, incorporating experience from the field, is therefore proposed. Starting with neutral earthing, only specific earth cables should be used until the weak spots are localized and rectified. It is useful to produce a lay-out diagram for the equipment showing earth, power, and data cables. At the same time, the measurement procedures should be defined and documented to ensure comparable quality for equipment modifications and extensions.

Ageing and wear

"Moisture, temperature fluctuations, coolant, solvent vapours, vibrations and alternating flexural loads continually affect the field bus installation over its entire life," explains Hans-Ludwig Göhringer. From the time of commissioning at the latest, these various influences leave their mark in the shape of wear on the bus installation. Without maintenance measures, sooner or later the signal-to-noise ratio is used up and the equipment stops.

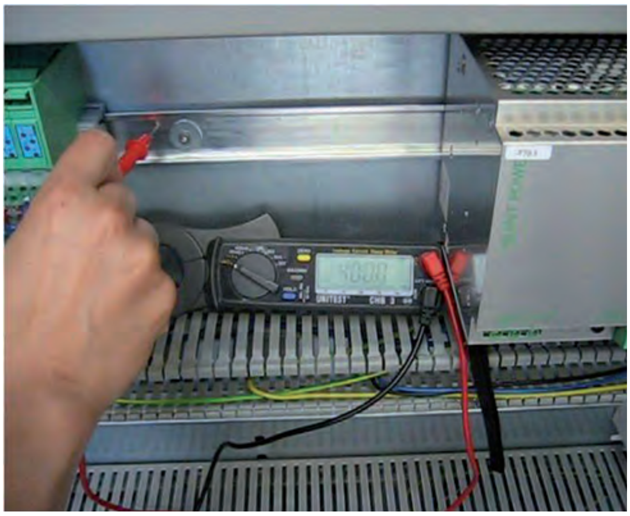


Figure 4: Detection of earthing contact problems caused by oxidation processes using a clip-on ammeter (Photo: IVG Göhringer)

Examples of ageing:

- Oxidation of contact surfaces
- Contacts being contaminated by dust, oil, adhesive and metal dust
- Cable failure in the cable track
- Cold soldering points caused by mechanical loading and strains associated with alternating temperatures
- Drying out of electrolytic capacitorsShort circuit caused by mechanical friction
- Loading of bus cable by chemicals and solvents
- Formation of whiskers on printed circuit boards
- Embrittlement of plastics due to UV radiation

The wear cannot be measured or predicted. Continual condition monitoring has proved a suitable strategy. Fieldbus systems like CAN are reliable systems with error tolerance thanks to their functional principle. Special mechanisms, such as automatically repeating telegrams, compensate for errors to a certain extent without the user even noticing. The CAN quick tester C-QT 15 from IVG Göhringer makes use of this system. The diagnosis module is attached at any point on the CAN network, where it works completely reactionless. It does not measure physical parameters such as voltage level or signal times; instead it records errors at protocol level. More specifically, the modules detect a deterioration in bus communication by detecting missing telegrams, repeat telegrams and missing communication partners. These errors are indicated by LED and using a potential-free alarm contact. The potential-free alarm contact of the C-QT 15 can trigger a warning light or siren to indicate an error. The alarm contact can also be analyzed by the superordinate controller or main computer.

Conclusion

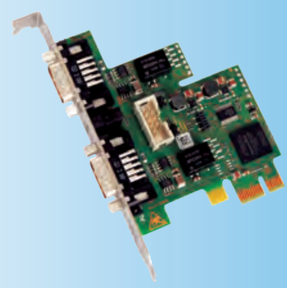
Although maintenance staff have increasingly focused on EMC in recent years, maintenance is often only deployed in the event of unexpected stoppages. However, the aim of efficient maintenance must be to maintain the performance of the bus systems and avoid unforeseeable faults. The CAN quick tester C-QT 15 of IVG Göhringer offers a simple solution. The compact diagnosis modules provide the user with continual monitoring. When the first telegram fails to appear, the maintenance staff can respond and scan the equipment for the error patterns described here. ◀

Author

Gerhard Bäurle
IVG Göhringer
info@i-v-g.de
www.i-v-g.de

www.esd.eu

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


CAN-PCIe/402

CAN-USB/400

CAN-PCI/400

Ethernet

CAN-USB/2

Gateways

All you CAN plug

CAN-PCIe/402

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

CAN-USB/400

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

CAN-PCI/400

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

CAN-PCI104/200

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

EtherCAN/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/S7-CAN

Operating Systems

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

CAN Tools

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

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

esd gmbh • Vahrenwalder Str. 207 • 30165 Hannover • Tel: (0511) 372 98-0 • Fax: (0511) 372 98-68 • info@esd.eu • www.esd.eu
esd electronics Inc. • 70 Federal Street • Suite 2 • Greenfield • MA 01301 USA • Phone: 413-7723170 • us-sales@esd-electronics.com • www.esd-electronics.us

Classic CAN
Improved CAN FD
Classic CAN
Solutions for
CAN