

December 2022

CAN Newsletter

Hardware + Software + Tools + Engineering



*The future of CiA – CAN XL ecosystem
and CiA profiles*

Non-automotive CAN applications

*Doctor CAN – embedded communication
in healthcare*

History and trends

www.can-newsletter.org



PCAN-Diag FD New J1939 Add-in

■ PCAN-Diag FD: CAN & CAN FD Diagnostic Device

The PCAN-Diag FD is a handheld device for the diagnosis of CAN and CAN FD buses at physical and protocol levels.

- High-speed CAN connection (ISO 11898-2)
 - Complies with CAN specifications 2.0 A/B and FD
 - CAN bus connection via D-Sub, 9-pin (CiA® 303-1)
 - Switchable CAN termination for the connected bus
- Power supply via rechargeable batteries or a supply unit
- Clear listing of the CAN traffic with various information
- Transmitting individual messages or CAN frame sequences
- Configurable, readable CAN ID and data representation
- Recording of incoming CAN messages
- Playback of trace files with optional loop function
- Measurement of the CAN bus load and termination
- Voltage check at the CAN connector for pins 6 and 9

Oscilloscope

- Function specially designed for CAN for a qualitative assessment of the signal course on the CAN bus
- Two independent measurement channels, each with a maximum sample rate of 100 MHz
- Display of the CAN-High and the CAN-Low signals as well as the difference of both signals
- Trigger configuration to various properties of CAN messages like frame start, CAN errors, or CAN ID

Now available with J1939 support

The new J1939 Add-in extends the functional range of the diagnostic device by the support for the SAE J1939 standard. The CAN data traffic is interpreted according to the included J1939 database and is represented in a way that is understandable for the user.

Features

- Representation of J1939 data interpreted according to PG and SP definitions
- SAE J1939 database with all definitions and the included parameters
- Decoding of multi-packet messages with payload data up to 1785 bytes
- Support for address claiming
- Display of DM and DTC diagnostic data

The J1939 Add-in is activated with a device-bound license which can also be purchased afterwards for a PCAN-Diag FD.



www.peak-system.com

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

PEAK-System Technik GmbH

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

PEAK
System



History and trends

The future of CiA – CAN XL ecosystem and CiA profiles	4
Non-automotive CAN applications	6
Doctor CAN – embedded communication in healthcare	10

Imprint

Publishing house
 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

Reiner Zitzmann (CEO)
 VAT-ID: DE812852184
 HRB: AG Nürnberg 24338

Publisher
 CAN in Automation e. V.
 Kontumazgarten 3
 DE-90429 Nuremberg
 VAT-ID: DE169332292
 VR: AG Nürnberg 200497

Editors
 Olga Fischer (of)
 Cindy Weissmueller (cw)
 (responsible according to the press law)
 Holger Zeltwanger (hz)
pr@can-cia.org

Layout
 Nickel Plankermann

Media consultants
 Tobias Kammerer
 Birgit Ruedel (responsible according to the press law)
publications@can-cia.org

Downloads September issue
 (retrieved November 24, 2022)
 6140 full magazine

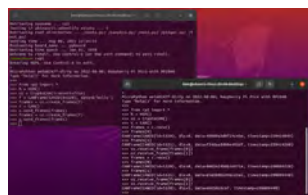
© **Copyright**
 CAN in Automation GmbH

The views expressed on CAN Newsletter magazine are not necessarily those of CiA e. V. While every effort is made to achieve total accuracy, neither CiA e. V. nor CiA GmbH can be held responsible for any errors or omissions.



Applications

CAN-based drive control for a robotic manipulator	20
Showing that electric aviation is possible and beyond	32
Converting mixed sensor data to CAN (FD)	34



Security

Securing CAN: Introduction to CryptoCAN	28
Mobile configuration, service, and diagnostic access to CAN systems	38



Transceiver

CAN transceiver fault detection with algorithm	16
The new dynamic parameters of CAN SIC	24



Brief news

Standards and specifications	22
------------------------------	----

Ending the 30-year anniversary

The 30th anniversary year of CAN in Automation (CiA) and the CAN Newsletter magazine comes to its end and so is the “history and trends” series. In all four issues of 2022, the CAN Newsletter magazine included feature articles which focused on dedicated application fields. These contained: ‘CAN in elevators’, ‘CAN in lower OSI layers’, ‘CAN in agriculture and farming’, ‘From classic CANopen to CANopen FD’, ‘CiA members from the beginning’, ‘CAN goes on and under the sea’, ‘CAN in air and space’, ‘CAN on construction sites’, ‘CAN on rails’, ‘CAN in healthcare’, ‘Non-automotive CAN applications’, and ‘The future of CiA’. All “history and trends” articles (and more) [can be downloaded and read here](#). As for the remaining days and weeks of 2022, the editors of the CAN Newsletter wish you a peaceful and happy end of the year and look forward to continue 2023 with you.

If you don't want to miss an issue of the CAN Newsletter magazine, you can subscribe to our monthly free-of-charge CAN Info Mail (CIM) email service which informs you on publication of the magazine and also provides other CAN-related news. To sign up for the CIM, send an e-mail to mail@can-cia.org. [Current issue](#).

The future of CiA – CAN XL ecosystem and CiA profiles

The 30th anniversary year comes to its end. Time to look to the future. CAN in Automation (CiA) is going to continue developing the CAN XL ecosystem. The nonprofit association is also committed to provide profile specifications to enable off-the-shelf interoperability between devices.

(Source: Adobe Stock)

Regarding lower CAN layers, CiA maintains all three generations: Classical CAN, CAN FD, and CAN XL – not forgetting the sideline CAN FD Light. CAN XL is an approach with data link layer add-on functions, such as CANsec, a security protocol, and frame fragmentation providing a quality-of-service (QoS) option. Frame fragmentation is suitable to decrease the network latency time needed when there are hard real-time requirements.

Additionally, the CAN XL protocol embeds besides DLL management features (e.g. priority ID) also higher OSI layer management features. One is the service data unit type (SDT). It is an 8-bit field, which indicates the used higher-layer protocol or the “tunneled” protocol (e.g. Classical CAN, CAN FD, or IEEE 802.3 MAC frames). The CiA 611-1 document defines the SDT codes and specifies the SDU (service data unit). But this is just the first piece of the CAN XL ecosystem. CiA develops a whole bunch of documents to support the usage of CAN XL and to simplify the CAN XL network design by means of recommendations, implementation and user guidelines as well as application notes. This will keep CiA members busy in the next years.

The adaptation of CAN XL by CANopen and J1939 is another topic, which CiA has on its to-do list. But currently, CiA struggles with the acceptance of CANopen FD. One of the hurdles is the design of longer networks. CAN FD is already rather successful in the passenger car business. But in applications requiring network length of 250 m and more, appropriate experiences are missing. The users – in particular, those with low-volume applications – have not the resources for the basic research. CiA is committed

to support them. In cooperation with researchers, CiA is going to set up an evaluation project for CAN FD communication in larger topologies. CiA members are also developing a CAN XL simulation approach, in order to check the feasibility of a CAN XL network design before building network prototypes.

To promote CAN XL, CiA members have established the [Marketing Group \(MG\) CAN XL](#). First actions include the organization of a CAN XL plugfest in Detroit area next year. Other marketing opportunities are application notes, demonstrators to be shown at trade shows, etc.

Interoperability is the objective

Compatibility is nice to have. It can be tested by means of conformance tests. However, conformance testing is the same as spelling and grammar checking in human communication. It increases the possibility of understandability (in technical communication: interoperability), but does not guarantee it. Even properly-spelled information can be mis-

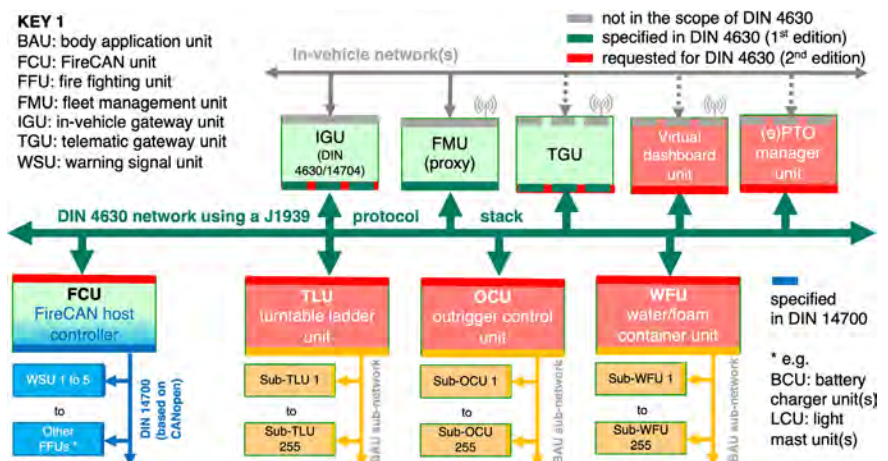


Figure 1: Example of a fire-fighting vehicle body application with multiple Classical CAN networks with different application layers (Source: CiA)

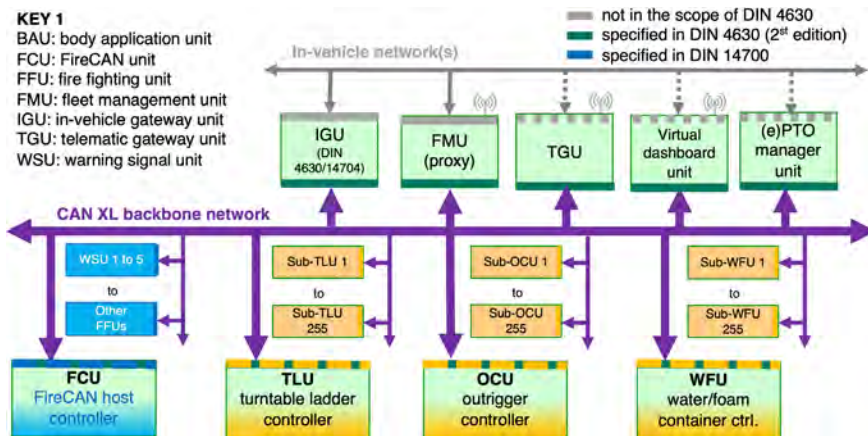


Figure 2: Example of a future fire-fighting vehicle body application using a single CAN XL backbone network implementing virtual networks (Source: CiA)

understood or misinterpreted. To prove the interoperability of implementations can be done by means of so-called plugfests or a “golden” system. Testing the ITU (implementation under test) in a “golden” system with already proven implementation is an option, CiA uses this approach for interoperability testing of classic CANopen devices.

Since its early days, CiA organizes plugfests to test the interoperability of devices and components. In such plugfests, several implementations are connected to one network. CiA had organized multiple CAN FD plugfests for CAN FD controllers. One of the most frequent plugfests are organized for [CANopen Lift \(CiA 417 series\)](#): often twice a year such test sessions are scheduled.

This year, CiA members checked the first [CAN XL protocol controller implementations and CAN SIC XL transceivers](#). CiA will continue to organize such CAN XL plugfests proving the interoperability of controller and transceiver implementations as well as topologies.

To test interoperability of devices on the application level, requires profile specifications. Profiles specify the content of messages. In CANopen and CANopen FD, this includes the object dictionary and PDO (Process Data Object) specification. CiA has developed many device, application, and interface profiles. CiA will continue this, enabling device suppliers to develop products with off-the-shelf plug-and-play capability. Not all existing CiA profiles have been upgraded for CANopen FD communication. There are still many CiA profiles limited to classic CANopen usage. The first CiA profiles have been adapted to J1939 meaning they support J1939-based application layers.

Device and network scalability is the goal

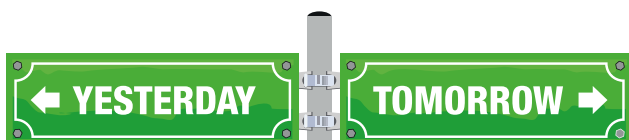
In some applications, multiple and different communication tasks are demanded: This can include remote control and remote diagnostics, local control, cloud communication (Internet of Things), etc. In order to provide network

scalability, logical units and logical controllers need to be specified. They can be mapped to different network architectures. CANopen and CANopen FD as well as CiA application profiles support such an approach. Typical examples are the CiA 417 application profile for elevator control systems and CiA 422 application profile for refuse collecting vehicles. Recently, CiA initiated the Special Interest Group (SIG) fire-fighting developing a framework for body applications on fire-fighting trucks. A similar approach is the development of application profiles for rolling stock.

In all these applications, CAN XL can overcome the limitations, when mapping them to Classical CAN or CAN FD networks. CAN XL provides with the 8-bit VCID (virtual CAN network identifier) field the possibility, to virtualize different networks on the same cable. This can save a lot of wiring and enables scalability of network architectures depending of the needed bandwidth. A side effect is the saving of CAN hardware interfaces as well as hardware bridges and switches.

With CAN XL, there are new design options to backbone multiple communication applications. This is already supported in classic CANopen, but with the limitation of an 8-byte data field and up to eight logical devices. One of the first approaches was the CiA 407 application profile for passenger information systems developed end of the 1990ties. This is internationally standardized in the EN 13149 series (CANopen-based passenger information system for public transportation). With the 2048-byte data field provided by CAN XL, you can implement a multi-PDU (protocol data unit) solution supporting such approaches much smarter as you could do with Classical CAN and CAN FD networks.

With these new features of the CAN XL data link layer in mind, CiA members can specify new virtual application profiles within the next years. Especially, applications with network scalability requirements can now be satisfied more easily. Interoperability of devices and scalability are two sides of the same coin, which can be achieved by means of CiA application profiles. The CiA Task Force CAN XL application, established recently, will provide success stories, how you can achieve this. ◀



Author



Holger Zeltwanger
CAN Newsletter
pr@can-cia.org
www.can-newsletter.org

History and trends:

Non-automotive CAN applications



(Source: Adobe Stock)

Although CAN was originally designed for passenger cars, non-automotive users applied CAN first. Of course, the volume is not that high as for road vehicles, but every little helps. Especially, when the little is widely spread.

The first non-automotive users included textile machines (Lindauer Dornier), elevators (Kone), [medical devices](#) (Phillips Medical Systems), and milking systems (Impulsa). CAN in Automation (CiA) founded in 1992 was focused in the beginning on industrial automation and machine embedded networks applying CAN networks. Already in 1992, CiA members developed the CAN Application Layer (CAL) specification (CiA 200 series), a pure OSI layer-7 approach (OSI: 7-layer open systems interconnection model). It was somehow heavy to implement eating a lot of resources. Based on CAL, Bosch and some partners developed in a European research project that what became a widely accepted application layer and communication profile. This is well known as CANopen (CiA 301). At the Hannover Fair 1994, CiA members exhibited a CANopen-based demonstrator. This was the starting point of the [CANopen success story](#). In 1999, CANopen was internationally standardized in EN 50325-4 thus accelerating a worldwide acceptance. In 2001, CiA has released the CiA 304 (later published as EN 50325-5) specifying functional safety for CANopen applications. Beside the base CANopen specification CiA 301, the CiA 3XX series comprise the CANopen-related additional features.

Standardized profiles simplify implementations

CANopen, originally designed for motion-oriented machine control systems, was and is often used as embedded network in all kinds of machines. Additionally, several compa-

nies provide IEC 61131-3 programmable host controllers with CANopen NMT (network management) manager functionality.

CANopen is a framework of specifications including interface, device, and application profiles as well as technical reports and recommendations. CANopen profiles specify required configuration parameters, process parameters, and diagnostic information for devices or whole applications in a standardized manner. Implementing a profile, device manufacturers may supply diverse markets with devices implementing the same profile-compliant electronic interface. System integrators can choose from devices implementing the same CANopen interface provided from different manufacturers. The usage of standardized CANopen interfaces allows setup, monitoring, analysis, and diagnostics of the networked devices using off-the-shelf tools. ▶



Figure 1: CANopen with its flexible configuration capabilities was originally designed for motion-oriented machine control systems (Source: CAN in Automation)



Figure 2: CAN and CANopen are well accepted in forklifts and pallet stackers (Source: Jungheinrich)

CiA members developed some generic device profiles, for example, [CiA 401](#) (modular I/O devices) and [CiA 402](#) (drive devices and motion controllers). CANopen I/O modules are available from Beckhoff, Peak-System Technik, Wago, Phoenix Contact, STW, Sontheim, Sys Tec Electronic, Inter Contol, B-Plus, Epec, ESD Electronics, Pixsys, Frenzel + Berg, Data Panel, Microcontrol, Axiomatic, EAO, and others.

The [CiA 402](#) series is a de-facto standard, which is also internationally standardized in IEC 61800-7-201/-301. It covers inverters, servo controllers, and stepper motors. This profile is also used by some other communication technologies with non-CAN physical layers. An incomplete list of companies implementing CiA 402 in their devices includes Schneider Electric, Maxon/Zub, Elmo Motion Control, JVL, Dunkermotoren, Nord Drivesystems, Harmonic Drive, Faulhaber, Advanced Motion Controls, Thomson Industries, Gefran, Linak, Applied Motion Products, Trinamic, Celera Motion, Ametek, LTI Motion, Wittenstein, Engel Elektroantriebe, Heason, Technosoft Motion, KEB, Axor, EBM-Papst, Copley Controls, Jenaer Antriebstechnik, Parker Hannifin, Servotronic, Nanotec, Jetter, Auxind, and Koco Motion. The most of them are (or have been) CiA members.

Since introduction of CAN FD in 2011, CiA members worked on adaptation of CANopen specifications to the higher throughput (more than 1 Mbit/s) and payload of

64 bytes (instead of 8 bytes). The CiA 1301 (CANopen FD application layer and communication profile) makes use of CAN FD's higher data throughput and keeps the key-attributes of CANopen. For drives, the CiA 402-6 specifies the CANopen FD PDO (process data object) mapping. This provides a simplified solution for synchronization of movements in several applications (e.g. movement start in multi-axis systems) as the control words for several axes are merged in one PDO. Thus, a single CAN FD data frame is utilized to transfer drive commands to all drives at the same time.

In its 30-years history, CiA has also released very special CANopen profiles. For example, the CiA 420 series for extruder downstream devices. In co-operation with the nonprofit Euromap organization, there have been specified profiles for saws, corrugators, and a second extruder. Other special profiles include those for measuring devices and closed-loop controllers ([CiA 404 series](#)) used for example in injection-molding machines. But not just the plastic-processing industry adapted CANopen. Printing machines (e.g. Heidelberger Druckmaschinen), weaving machines, and other machinery use embedded CANopen networks.

Other non-automotive markets

CANopen is not limited to industrial automation. It is used in many other industries. One of the success stories is [CANopen Lift](#) (CiA 417 series). In the beginning, there were just a few companies developing the application profile for elevator control systems. Nowadays, CiA organizes biannually plugfests to test the interoperability of CANopen Lift controllers and units. This application profile is the only standardized network approach for elevators connecting the host controller with the car drive, car doors, and panels as well as tableaux.

In construction machines, such as vehicle-mounted telescope cranes and earth moving equipment, and mining machines, CANopen is on duty. CANopen is also well accepted in other kinds of "machines on wheels". These include forklifts, straddle carriers, and automatic guided vehicles (AGV). Some examples are [Smartfork from Vetter](#), [RFID reader from Idtronic](#), and [GLS100 line guidance sensor by Sick](#). CiA member Jungheinrich uses CANopen ▶

CiA 402 dominated at SPS 2022

(Source: Adobe Stock)

On the SPS 2022 a lot of drives and motion controller suppliers have shown their CiA 402 compliant products. Although not all companies did promote visibly their CANopen interfaces, many of them sell a lot of these. Read more in the [CAN Newsletter Online](#).



Figure 3: CiA has developed several profiles dealing with battery and energy management. These will be discussed in a [web-based workshop](#) in December 2022. (Source: Adobe Stock)

also in its [pedestrian stackers](#). In many of these “machine-on-wheels” networks, CANopen encoders ([CiA 406](#)) and inclinometers ([CiA 410](#)) are applied. Additionally, there are hydraulic devices ([CiA 408](#)) connected in conjunction with linear encoders.

There is also another kind of “machines on wheels”, which uses CAN-based networks with standardized application layers and profiles (CANopen as well as J1939). These are truck and trailer body applications. Some are simple, while others are more complex comprising multiple CAN segments running even different higher-layer protocols. Refuse-collecting trucks are an example, which is

using the CleANopen application profile ([CiA 422 series](#)). In fire-fighting trucks some equipment is linked by a CANopen-based profile (DIN 14700). The CiA 413 series specifies the CANopen gateways to J1939-based in-vehicle networks. The latter can use such application profiles as SAE J1939-71, ISO 11992-2, -3, and -4 for truck/trailer communication, ISO 11783-based (Isobus), NMEA 2000, etc.

CiA has also developed a bunch of profiles for [rolling stock](#). There are commuter trains, diesel locomotives, trams, and metros implementing CANopen networks to control the rail vehicles. Recently, CiA received a request to continue the development of an application profile for HVACs (heating, ventilation, and air-conditioning). Such a profile would not be limited to locomotives and coaches.

In this-year, the “history and trends” article series CAN Newsletter additionally reported about CAN use in such non-automotive applications as [agriculture and farming](#), [maritime electronics](#), [air and space](#), [construction applications](#), [elevators](#), [healthcare](#), as well as [rails](#).

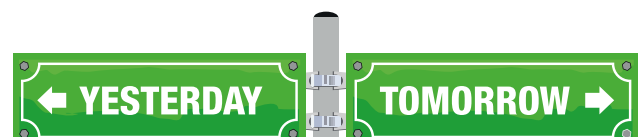
CANopen device profiles

- ◆ CiA 401 for modular I/O devices
- ◆ CiA 402 for drives and motion control
- ◆ CiA 404 for measuring devices and closed-loop controllers
- ◆ CiA 406 for encoders
- ◆ CiA 408 for fluid power technology
- ◆ CiA 410 for inclinometer
- ◆ CiA 412 for medical devices
- ◆ CiA 413 for truck gateways
- ◆ CiA 414 for weaving machines
- ◆ CiA 418 for battery modules
- ◆ CiA 419 for battery chargers
- ◆ CiA 420 for extruder downstream devices
- ◆ CiA 434 for laboratory automation systems
- ◆ CiA 442 for IEC 61915-2 compatible motor starters
- ◆ CiA 443 for SIIS level-2 devices
- ◆ CiA 444 for container-handling machine add-on devices
- ◆ CiA 450 for pumps
- ◆ CiA 452 for PLCopen motion control
- ◆ CiA 453 for power supplies
- ◆ CiA 458 for energy measurements
- ◆ CiA 459 for on-board weighing devices
- ◆ CiA 460 for service robot control systems
- ◆ CiA 461 for weighing devices
- ◆ CiA 462 for item detection devices
- ◆ CiA 463 for IO-Link gateway

Outlook

Adaptation of CiA profiles to CANopen FD is ongoing, but industrial migration to CANopen FD is still in its infancy. Some of the above-mentioned application fields are conservative. They need a long time to adapt improved CAN FD networks. SAE has done its homework regarding J1939: Since 2020, the CAN FD specifications (J1939-22 and J1939-17) are published. CAN FD would allow to implement functional safety or cybersecurity due its 64-byte data-field and its higher throughput (when running 2 Mbit/s in the dataphase). Certification of CANopen FD devices is already possible by CAN in Automation.

The success of CANopen relates to the wide range of generic and specific profiles. Recently, CiA has started to adapt its encoder and inclinometer profiles to J1939 (CiA 406-J and CiA 410-J). Other profiles will follow. ▶





(Source: Adobe Stock)

- ◆ CiA 415 for sensor systems in road-construction and earth-moving machines
- ◆ CiA 416 for building door control
- ◆ CiA 417 for lift control systems
- ◆ CiA 421, CiA 423, CiA 424, CiA 426, CiA 430, CiA 433 for rail vehicles
- ◆ CiA 422 for municipal vehicles
- ◆ CiA 425 for medical diagnostic add-on modules
- ◆ CiA 436 for construction machines
- ◆ CiA 437 for grid-based photovoltaic systems
- ◆ CiA 447 for special-purpose car add-on devices
- ◆ CiA 454 for energy management systems
- ◆ CiA 455 for drilling machines

There is already the third CAN generation coming: CAN XL with payloads up to 2048 byte and bit rates above 10 Mbit/s. With CAN XL you can run heterogenous higher-layer protocols on the same cable. This would simplify the effort on wiring harnesses and reducing the number of CAN hardware interfaces, because CAN XL enables virtual networks. CAN XL is intended for backbone and sub-backbone network applications. It also enables an easy integration into TCP/IP network systems. ◀

Autor

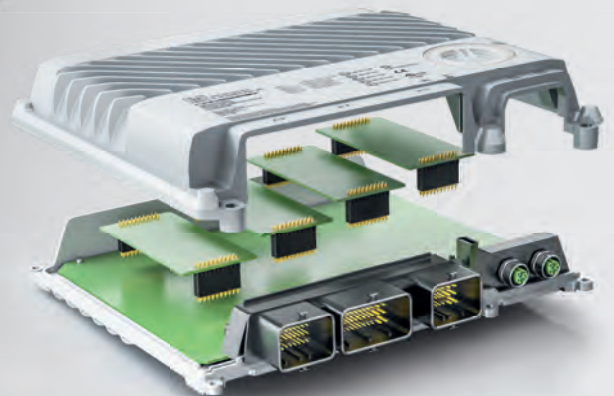


Olga Fischer
 CAN Newsletter
pr@can-cia.org
www.can-newsletter.org



Completely integrated automation for mobile machinery - X90

Complete portfolio:
www.br-automation.com/mobile-automation



- Easy handling
- Integrated safety
- Faster development



PERFECTION IN AUTOMATION
 A MEMBER OF THE ABB GROUP



History and trends:

Doctor CAN – embedded communication in healthcare

Already in the early days of CAN, it was used in medical devices as embedded network. Still nowadays, CAN is because of its reliability and robustness a preferred network technology in healthcare.

(Source: Adobe Stock)

CAN has been used in medical devices since a long time. Today, CAN networks are used also in intensive care units including patient beds, in operating rooms, and in other healthcare equipment. Most of these CAN networks are embedded or even deeply embedded. Open networks are used for example to connect medical imaging systems to contrast media injectors.

CAN is used in medical devices such as X-ray machines, magnetic resonators, angiographs, computer tomographs, and others. These devices may implement cascaded CAN networks for embedded and deeply embedded control applications. Sub-systems with a standardized CANopen interface include collimators and dose-meters. Also, medical imaging devices may be connected to contrast media injectors by an open CANopen-based network. CAN networks can also connect all devices and units inside operating rooms to enable fast and monitored plugging together of operating gear so as to avoid any omissions and to check on the functionality of all devices. Intensive care units (ICU) are another use case for CAN in healthcare. CAN is used as deeply embedded network for internal control purposes. The interconnection of ICUs via

CAN networks is a further application area. Some sophisticated patient beds use an embedded CAN control system for the motion controllers and the different user interfaces. The beds additionally provide a CAN interface to connect for example a blood-pressure monitor.

Philips Medical Systems was one of the early adopters of CAN communication in X-ray devices, computer tomography, and other medical devices. As an early CAN in Automation (CiA) member, the healthcare company supported the development of the CAN Application Layer (CAL) released by CiA in 1993. It was a pure application layer (layer-7) approach, which was the predecessor of the CANopen application layer and communication profile. In 1993, Siemens implemented CAN networks with proprietary higher-layer protocols in its computer tomography systems.

CAL was also used by Karl Storz, a Swiss company, for its endoscopy devices. Mid of the 90ties, endoscopy pioneer Richard Wolf, a German company, connected its products via embedded CANopen networks. Additionally, the company linked operating (OR) tables, surgical lights, and other devices from third-party suppliers by means of a second CAN interface.

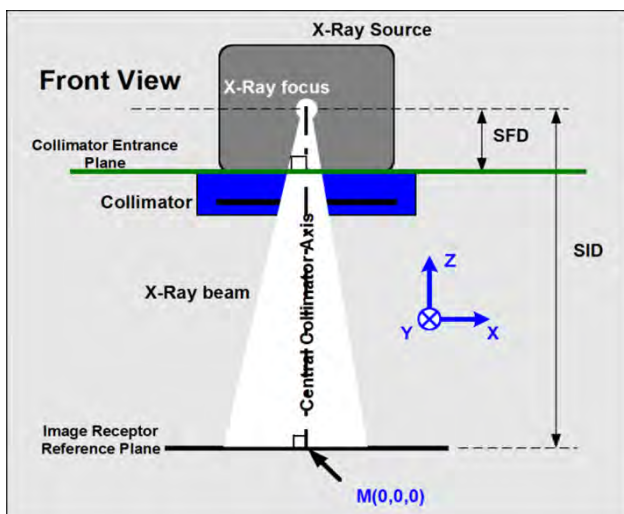


Figure 1: Collimator coordinate system, whereby the individual coordinates are as seen from a front view (Source: CiA)

CANopen profiles for medical devices

With the introduction of CANopen, Siemens (nowadays Siemens Healthineers) and GE Healthcare migrated from proprietary CAN-based embedded networks to this application layer. CiA members developed CANopen profiles for automatic X-ray collimators (CiA 412-2) and dose measurement systems (CiA 412-6).

The CiA 412 CANopen profiles for medical devices specify general definitions (Part 1), the CANopen interface for automatic X-ray collimators (Part 2) as well as for dose measurement systems (Part 6). Using standardized CANopen interfaces, device manufacturers may supply diverse markets with medical devices implementing the same electronic interface according to CiA 412 and can simply vary the appropriate application software. A system designer may choose between CANopen devices from different manufacturers implementing the same profile-compliant ▶

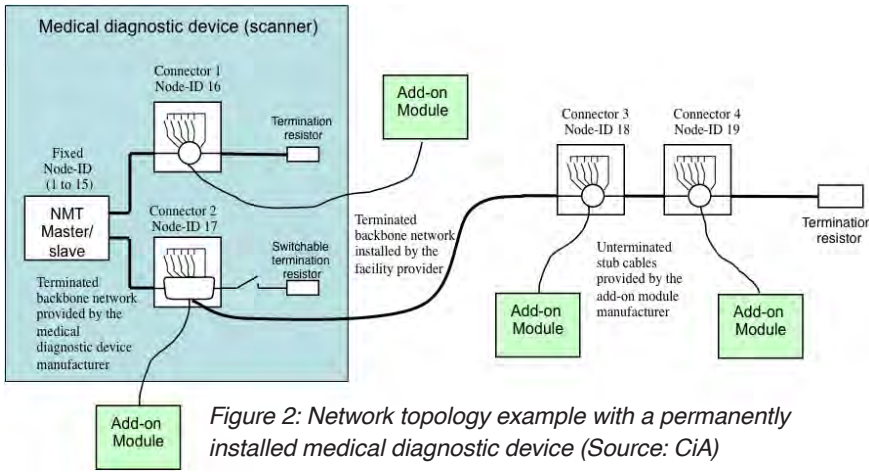


Figure 2: Network topology example with a permanently installed medical diagnostic device (Source: CiA)

functionality. For development, analysis, and maintenance of the devices, CANopen tools can be used.

The CiA 412-2 document for automatic X-ray collimators, represents the CANopen device profile for generic X-ray collimators, and as such describes the generic subset of collimator functionality. A collimator has three basic functions for which the profile specifies the appropriate configuration, application, and diagnostic parameters. The main function limits (or collimates) the X-ray beam (e.g. rectangular collimation) issued by an X-ray emitting source (X-ray tube) to a defined (receptor) format. Filters may be applied to influence spectral characteristics of the X-ray beam. The visual simulation of the X-ray beam is the third specified functionality. Some automatic X-ray collimators support local control functionality. The defined collimator functionality coordinates (X, Y, s, ω, D) may be controlled either in position or velocity mode. Devices compliant to this profile are required to support the emergency message (CiA 301). The defined device errors are sorted in warnings, recoverable errors, and non-recoverable errors.

The introduced collimator device FSA (finite state automaton) specifies the application behavior as well as the corresponding state transitions of the collimators. As a collimator is usable with local control even when the CAN network does not work properly, the communication FSA (CANopen NMT server FSA, CiA 301) and the collimator FSA are very loosely coupled. Also defined is a coordinate FSA applicable for the symmetric rectangular collimation sets, the quadrangle collimation sets, the circular collimation sets, as well as the spatial filter sets. The third specified FSA (homogeneous filter FSA) has the same states as the coordinate FSA with a different definition for some states. In addition, the X-ray visualization FSA is defined. Further, the profile provides some use case scenarios e.g. coordinate motion between the defined limits, changes of the SID (source image distance) value, etc. The CiA 412-2 pre-defines one RPDO containing the collimator command and the target x-y-position value as well as one TPDO providing the collimator state and the actual x-y-position value.

The CANopen dose measurement system (CiA 412-6) measures the X-ray dose and the dose area product. In addition, the dose area product rate, dose rate, RD (reference distance) entrance/skin dose, RD entrance/skin dose rate, MD (measured distance) entrance/skin dose, MD entrance/skin dose rate, irradiation time, chamber

temperature, as well as the air pressure values are measured. The actual measured values (called field values) are converted to values with a real physical dimension (called process values). The profile specifies all required objects to fulfill this conversation and to represent the mentioned values in a standardized manner. Additionally, CiA 412-6 introduces an FSA for the dose measurement systems. The profile defines one RPDO and two TPDOs respectively transferring the control word (RPDO1) and the status word

(TPDO1) as well as the current process value (TPDO2). Profiles for X-ray generators (Part 3), patient tables (Part 4), and X-ray stands (Part 5) are also intended in the future.

Nowadays, many of the medical device suppliers use CANopen as embedded network for different purposes in X-ray machines, in computer tomography, and angiography. This includes United-Imaging Healthcare, a Chinese CiA member, which has equipped its products with CANopen networks to integrate devices from several suppliers, especially motion controllers and I/O modules.

CANopen for medical diagnostic add-on modules

Some medical scanner devices need to communicate with contrast media injectors. For this purpose, CiA members specified around 2011 the CiA 425-2 profile. The SIG (special interest group) contrast media injector is still adding new features and updates. The scope of the group is the enhancement and maintenance of profiles for medical add-on devices such as contrast media injector.

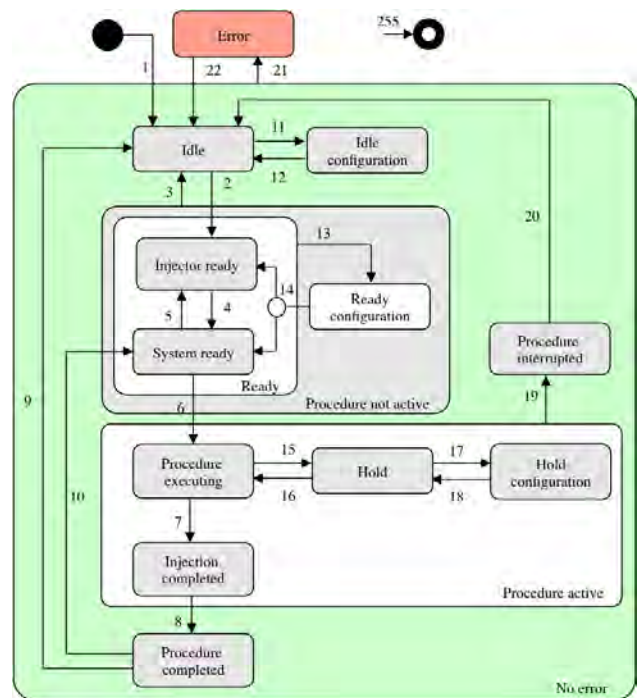


Figure 3: Injector state machine; FSA (finite state automaton) (Source: CiA)



Figure 4: The Econ100 from HMS/lxxat used for mammography (Source: IMS/HMS/lxxat)

The CiA 425 profile specifies the CANopen communication between medical sub-systems (diagnostic devices) and their add-on modules. It covers the general definitions (Part 1) and the CANopen interface for injector devices (Part 2). Injectors from different manufacturers implementing this profile may be attached to a diagnostic device via a standardized CANopen interface. Thus, a diagnostic device manufacturer does not need to hold several spare injectors available. The same injector interface implementation may be also used in diverse diagnostic devices of one manufacturer.

The physical device interface complies with IEC 60601-1 for medical electrical equipment. CANopen-related physical layer accords to CiA 301 version 4.2.0. Bit rate of 250 kbit/s is supported. Four connectors are specified. Five of the connector pins are used for geographical node-ID assignment. Medical diagnostic devices use the node-IDs 1 to 15. Geographical addressing is not mandatory for permanently connected nodes. Here the node-ID setup is internally done by hardware or software. The network installer has to ensure, that a node-ID is not used twice. Typically, the medical diagnostic device provides the CANopen NMT (network management) manager capability and the add-on modules provide the NMT server functionality.

The CANopen injector interface (CiA 425-2) specification is used to connect automatic and semi-automatic injectors to the CANopen network allowing operation



Figure 5: The Accutron CT-D CT861-2 double-head injector with a support arm (Source: Medtron)



Figure 6: With its surgical microscope with integrated OCT camera, Zeiss gives surgeons better insights into the transparent structures of the eye during surgery. A CAN interface is used in the panel PC. (Source: Zeiss)

of third-party products. It covers injectors connected to such medical diagnostic systems as angiography, computer tomography, magneto resonance, ultra-sonic, etc. The injector may provide up to eight configurable pistons. Each injector implements a certain compatibility class (0 to 5) showing the diagnostic device's capability to control the injector. This reaches from injector monitoring (class 0) to real-time adjustment of the injector parameters. The higher class provides the functionality of the lower class. If required, the injector may use the safety-related communication according to EN 50325-5.

The specified state machine (FSA) defines injector's application behavior. It is also specified on which events certain FSA transitions and actions are executed. The injector may be operated by local commands (not specified) or via CANopen network by the control word sent from the medical diagnostic device. The injector reports its state in the status word. The FSA is also controlled by detected errors.

The programmable injection protocol is a sequence of configured phases (injection, test bolus, delay, and wait) with defined actions. The injection and test bolus phases require configuration of total volume, total flow rate, and piston ratio. Optionally, the rise time and pressure limit may be configured. The diagnostic device controls and monitors the phase processing by means of the control word respectively the status word. The injector may support three operation modes (monitor, tracking, and control). The CiA 425-2 further defines the RPDO 1 and TPDOs 1 to 4, the complex data types, and the required application objects. Additionally, to the mentioned above, the latter include the current and configured values, physical units (specified according to CiA 303-2) and limitations for volume, pressure, and flow rate related to the corresponding phase types. Time-related values, piston attributes, configured language, supported CiA profile version, as well as the device's (diagnostic device and injector) description are specified within certain objects. The CAN Newsletter magazine published several articles regarding injectors:

- ◆ [CANopen object dictionary of an injector \(CAN Newsletter 1/2021\)](#)
- ◆ [Injector and scanner communication \(CAN Newsletter 4/2020\)](#)
- ◆ [Implementing a CANopen injector FSA \(CAN Newsletter 3/2020\)](#)

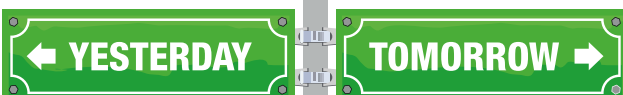


Figure 7: CAN is also used in dentist instruments as well as in dentist chairs to move the patient in position. (Source: Adobe Stock)

Application examples

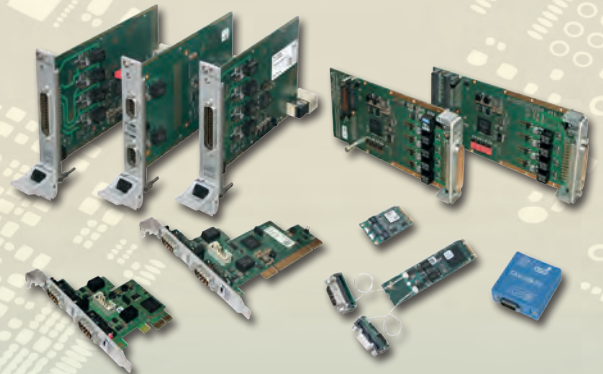
Already in 2005, the CAN Newsletter magazine reported about Medtron implementing CANopen in its contrast media injectors. In 2021, the CAN Newsletter Online [published an article](#) about CANopen-based CT imaging. Medtron and Siemens designed the Accutron CT-D CT861-2 injector communicating with Siemens Somatom go CT scanners via a CANopen interface. The double-head injector is needed for clinical CT (computer tomography) imaging procedures. It is mountable on the gantry by means of a Siemens support arm in which the power supply cable is routed. The company's CANopen interface Class IV allows a wireless synchronization of the injector and the scanner system to achieve a workflow. By mounting the injector onto the gantry, it is available next to the CT scanner table. This integration also enables a positioning closer to the patient. Medtron works in the field of medical engineering. It manufactures contrast medium injectors used across the globe within the imaging systems, such as magnetic resonance imaging, computed tomography, and angiography. The German company is also contributing to the development of the CiA 425-2 profile specification for CANopen injectors.

In 2018, the [CAN Newsletter magazine reported](#) about CAN devices from HMS/lxxat which were used in mammography. MS used the lxxat Econ100 embedded controller in their Giotto Class mammography machine. The machine can move around the patient taking X-ray photos from several different positions, providing physicians with better pictures for detecting breast cancer. The moving parts of the Giotto Class are mainly controlled using the CANopen protocol. The Econ100 manages the internal communication network and the logic control unit for about twenty different electronic boards. It controls movement, X-ray emission, data acquisition, visualization, and safety chain inside the machine. The controller features a Xilinx Zynq SoC – dual-core Cortex A9 processor as well as two CANopen ports which make it possible to configure ▶



www.esd.eu

All you CAN plug



CANopen^{FD}

CAN^{FD}

CAN / CAN FD Interfaces

Product Line 402 with Highspeed FPGA

- Various Form Factors**
 PCI, M.2, PCI Express[®] Mini, PCI Express[®], CompactPCI[®], CompactPCI[®] serial, XMC/PMC, USB, etc.
- Highspeed FPGA Design**
 esdACC: most modern FPGA CAN-Controller for up to 4 channels with DMA
- Protocol Stacks**
 CANopen[®], J1939 and ARINC 825
- Software Driver Support**
 Windows[®], Linux[®], optional Realtime OS: QNX[®], RTX, VxWorks[®], etc.



esd electronics gmbh
 Vahrenwalder Straße 207
 D-30165 Hannover
 Tel.: +49(0)511 372 98-0
 info@esd.eu | www.esd.eu

Quality Products -
 Made in Germany

esd electronics, Inc.
 70 Federal Street - Suite #5
 Greenfield, MA 01301
 Phone: +1 413-772-3170
 www.esd-electronics.us



Figure 8: The cough simulator in action (Photo: Kvaser)

communication at two different bit rates - to adjust to different stub lengths within the network. It offers two independent CAN networks with CANopen as higher-layer protocol for communication.

The Callisto Eye from Zeiss also uses a CAN interface. The panel PC is part of the Opmi Lumera 700 from Zeiss, which is an operating microscope suited for every ophthalmic surgery specialty. The assistance functions of Callisto Eye are surgeon-controlled – with either the foot control panel or handgrips.

Also in many dentist chairs, CAN-based motion controllers are applied to move the patient in position. Additionally, the dentist instruments are controlled via additional CAN networks. For example, Austrian CiA member, W&H Dentaltechnik, has equipped a range of its dentist instruments with CANopen interfaces. Such CANopen interfaces are not yet standardized by means of CiA profiles. The mechanical interfaces are already internationally standardized.

Coughing with CAN: In 2017, Niosh (USA) has [developed a cough simulator](#) to study how diseases like influenza can spread via airborne droplets. This simulator uses Kvaser's CAN-based Leaf Light HS v2 interface (Figure 8). It is a high-speed USB interface for CAN that offers loss free transmission and reception of basic and extended CAN frames on the CAN network. In this application case it connects the motor and the computer, transmitting

control commands from a custom-built National Instrument's Labview program, and sending back motor position feedback. The list on applications of CAN in healthcare is long and impossible to mention all. CAN is not often visible, but there in so many fields. ◀

Covid-19 and CAN



The coronavirus still goes around the world. The CAN networks used in medical equipment help indirectly in the fight against it. This article in the CAN Newsletter Online collected products and developments helping.



The disinfection robots by UVD robots (Denmark) are deployed in hospitals to disinfect rooms and equipment such as patient beds. They also use embedded CAN networks.

Author

Cindy Weissmueller
CAN Newsletter
pr@can-cia.org
www.can-newsletter.org





neoVI PI

Robust and Open Raspberry PI 4 Platform for CAN/CANFD Applications

Introducing the Automotive Industry's first open and robust platform for the Raspberry Pi. The neoVI PI has a built-in Raspberry Pi 4 Compute Module (RPi4 CM) that contains quad 64-bit processors and a gigabit Ethernet port, paired with Intrepid's CAN FD technology. This allows you to simulate, test and datalog with the flexibility that the Raspberry Pi 4 Compute allows.

The neoVI Pi has all the features of the RPi4 CM plus up to four CAN FD networks. The neoVI PI is designed and tested for the automotive environment. This includes a wide power supply range, EMC protection, rugged packaging and environmental testing. The neoVI PI allows you to use the Raspberry Pi 4 Compute while avoiding additional development to adapt to the automotive network environment. That makes the neoVI PI powerful enough to solve your vehicle network problems, yet small enough to fit in your backpack.

FEATURES

- Built-in RPi4 Compute Module supports all variations of EMCC, SDCard, and Wireless
- 2x internal ValueCAN4-2 for 4 CAN FD / CAN 2.0 Channels
- Intrepid Open source APIs on github/intrepidcs: libicsneo for C/C++ and python_ics for Python
- Automotive Power Supply (5-60V operation)
- 1x Native 1000BASE-T Ethernet with PoE sink support
- 4x High Speed USB Host Ports with high current sourcing
- Integrated Raspberry Pi Pico Module connected to RPi4 via USB
- M.2 NVMe slot for hosting PCIe flash up to 4TB
- Expandable IO : Internal RPi and Rpi Pico GPIO access with open connector pins for custom hardware applications
- Tested and Packaged for in-vehicle use
- HDMI connector for RPi4 OS display



For more information:



INTREPID
CONTROL SYSTEMS
www.intrepidcs.com

CAN transceiver fault detection with algorithm

This article discusses the fault detection feature of the MAX33011E CAN transceiver from Maxim Integrated (now part of Analog Devices). It also demonstrates how to implement the fault detection algorithm in firmware with example codes.

(Source: Adobe Stock)

Troubleshooting a CAN network when data is not able to be transmitted or received can be frustrating. Maxim has developed a built-in fault detection mechanism in the CAN transceiver that helps users to quickly identify the root cause.

Fault detection circuit function

The MAX33011E CAN transceiver requires 100 rising signal edges on the TXD pin (typically a few CAN frames) to enable the fault detection circuit. After the fault detection circuit is enabled, the transceiver can still transmit frames as normal.

When a fault condition is detected, the transmitter will be disabled and the Fault pin will be pulled to high through

the external pull-up resistor. When the system controller receives the Fault-pin signal, 16 low-to-high transitions on the TXD pin are required to shift out the fault code as shown in Table 1. Additional 10 low-to-high transitions on the TXD pin clear the fault and disable the fault detection circuit. For example, the overcurrent fault code is 101010 and its timing diagram is shown in Figure 1.

Fault conditions

The MAX33011E is the first CAN transceiver with a built-in fault detection circuit, claims the company. When the fault detection circuit is enabled, it can detect three types of common fault conditions (overvoltage, overcurrent, and transmission failure) on a CAN bus line as listed in Table 1.

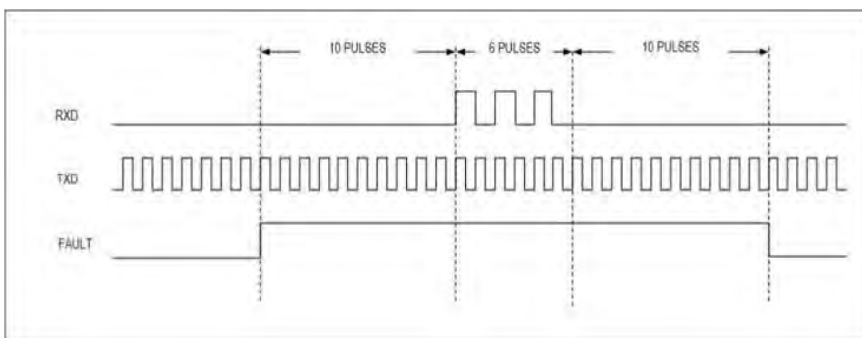


Figure 1: Overcurrent fault reporting timing diagram (Source: Maxim Integrated)

Overcurrent fault

Overcurrent fault is detected when the source current of CANH and the sink current of CANL are both higher than 85 mA (typically). The more probable cause of the fault is that CANH and CANL are shorted on the bus line. However, if the short is far away from the CAN node, it may not be detected due to a high cable impedance. Slowing down the CAN

Table 1: Three types of common fault conditions, which can be detected by the fault detection circuit (Source: Maxim Integrated)

Fault	Condition (Fault Detection Enabled)	Fault Code	Possible Cause
Overcurrent	CANH output current and CANL input current are both > 85 mA	101010	<ul style="list-style-type: none"> CANH shorted to CANL CANH connected to GND and CANL connected to V_{DD}
Overvoltage	CANH > +29 V or CANL < -29 V	101100	<ul style="list-style-type: none"> CMR fault
Transmission Failure	RXD unchanged for 10 consecutive TXD pulses, recommended minimum frequency = 200 kHz	110010	<ul style="list-style-type: none"> Open load (both termination resistors missing) on CANH and CANL Exceeds driver's common-mode range CANH and /or CANL connected to a fixed voltage source

signal frequency could lower the cable impedance and help to detect the short from a further distance. But, if the total resistance of the cable becomes significantly high, a short will not be detected even when the CAN signal is constantly in a dominant state. Figure 2 shows the maximum operating frequency for overcurrent detection versus cable length as a reference. A Cat5E copper clad aluminum cable is used. The maximum frequency will vary with the type of cable.

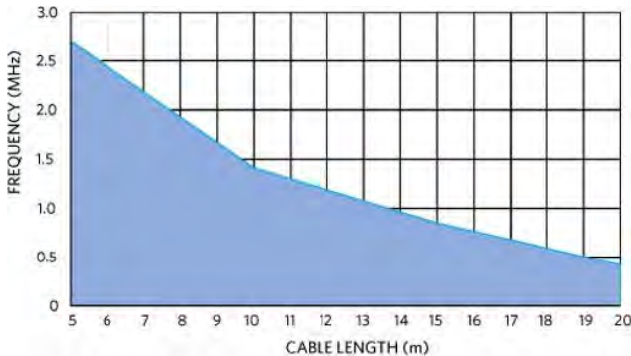


Figure 2: Maximum operating frequency for overcurrent detection vs. cable length (Source: Maxim Integrated)

Overvoltage fault

The MAX33011E common-mode input range (CMR) is ± 25 V. An overvoltage fault is detected when either the voltage on CANH is higher than 29 V or at CANL is lower than -29 V. This is caused by the CMR going beyond the specified value.

Transmission failure fault

After the fault detection circuit is enabled, the transceiver can still transmit CAN frames. In a normal operating condition, RXD signal echoes the TXD signal. In case the RXD signal does not echo the TXD signal for 10 consecutive pulses, the fault detection circuit generates the transmission failure fault. There are a few common possible causes that make RXD signal unable to echo the TXD signal:

1. If CANH and CANL are shorted to a supply and the transceiver is not able to overdrive the supply, the receiver will always see a fixed signal on the CAN bus line.
2. When the common-mode voltage exceeds the transceiver's common-mode range (-5 V to +10 V), the transceiver is turned off. When the transceiver is off, the CANH/CANL output will not reflect the signal on the TXD pin, and the receiver will see a fixed signal on the CAN bus line.
3. If no termination resistor is connected to the CAN node, it may cause a transmission failure. The termination resistors play a very important role of bringing CANH and CANL to the same voltage level in the recessive mode.

Without the termination resistors, the transceiver's internal common-mode voltage buffer can still bring CANH and CANL together, but at a much slower rate. The capacitive load on the bus line could also slow down CANH and CANL voltages from merging. When the controller sends pulses to the TXD pin, and if the recessive interval is not long enough for the differential voltage (CANH - CANL) to go below the input low-threshold for 10 consecutive pulse cycles (RXD signal stays low for the 10 TXD-signal pulses), a transmission failure fault will be reported. This also means if the TXD-signal high-time is too long, the CAN-bus-line signal could enter the recessive mode and the RXD signal will become high, no transmission failure fault will be reported. The recommended minimum TXD pulse frequency to detect a transmission-failure fault, is 200 kHz.

Fault detection algorithm

Maxim has developed an algorithm that can detect fault conditions on the CAN network using the MAX33011E transceiver, without interrupting the normal CAN communication. The following example Mbed codes were developed for the Nucleo-F303K8 platform by ST Microelectronics.

Typically, a micro-controller with a CAN peripheral is used in every node on a CAN network. To perform fault detection, the TXD and RXD pins of the micro-controller must be configured as GPIOs (general purpose input output) to bit-bang the TXD signal and to read the fault code from the RXD pin. An interrupt pin is needed to connect to the fault signal of the MAX33011E.

To avoid interruption of the normal communication, the algorithm needs a strong indication of a communication failure before entering the fault detection mode. The algorithm will enter the fault detection mode, if one of the following happens:

- ◆ Fault-pin signal goes high,
- ◆ Transmitter generates a bit-error frame,
- ◆ Transmitter error counter rises above 255 and the node enters the bus-off state.

The example code in Figure 3 configures the micro-controller pins as a GPIO when any of the above-mentioned conditions becomes true. In this example, the STM32F303K8 MCU (micro-controller unit) was used as the CAN controller. The example reference code was developed on [Mbed-OS](#), which is a free open-source embedded operating system. Mbed offers APIs (application programming interface) to configure the micro-controller's I/Os as digital input/output pins. Any other similar low-level APIs may also be used.

In the fault detection mode, the 2- μ s-TXD positive pulses should be used. After the positive pulse, the TXD signal can stay low as long as needed to process the algorithm. This allows the fault detection circuit to reliably

```

DigitalOut txd (PA_12); // Configures PA_12 of MCU (TXD of CAN controller) as digital o/p pin
DigitalIn rxd (PA_11); // Configures PA_11 of MCU (RXD of CAN controller) as digital i/p pin

```

Figure 3: The example code configures the micro-controller pins as a GPIO when any of the above-mentioned conditions becomes true (Source: Maxim Integrated)

```

static TIM_HandleTypeDef s_TimerInstance = {                               //Creates a timer instance (TIM2)
    . Instance = TIM2
};
__HAL_RCC_TIM2_CLK_ENABLE ();                                           //Enable TIM2 APB clock (72MHz)
s_TimerInstance.Init.Period.Prescaler = 16;                             //Set the counter prescaler value
s_TimerInstance.Init.CounterMode = TIM_COUNTERMODE_UP;                  // Set the counter in "UP" mode
s_TimerInstance.Init.Period = 500;                                     //Set the counter period
s_TimerInstance.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;          // Set the clock divisor value to none
s_TimerInstance.Init.RepetitionCounter = 0;                           //Disable the auto-reload
HAL_TIM_Base_Init(&s_TimerInstance);                                   // Initializes TIM base unit according to specified parameters
HAL_TIM_Base_Start(&s_TimerInstance);                                  //Start the timer in time-base mode

```

Figure 4: Example code to set up the timer (Source: Maxim Integrated)

```

InterruptIn fault (PA_0);                                               //Configure Fault pin as interrupt pin
fault.rise(&fault_init);                                               // Attach an interrupt callback function when fault pin goes high
h
int state=0;                                                            // This is the state of state machine for fault detection
void fault_init()
{
    fault.rise (NULL);                                                 //Detach an interrupt till state machine gets completed
    toggle_txd_i_ticker.attach_us(&toggle_txd_i,6); // Initiate a state machine to detect a fault, each cycle is 6us
}
void toggle_txd_i()
{
    DigitalOut txd(PA_12);                                             //Configure CAN TXD pin as digital o/p
    DigitalIn rxd(PA_11);                                             // Configure CAN RXD pin as digital i/p
    DigitalIn fault_pin(PA_0);                                         //Configure fault pin as digital i/p
    int status = fault_pin.read();                                     //Fault pin status is stored in status variable
    static int count,N;

    do {                                                                //This code toggles TXD for 2us duration using TIM2 tim
er
        txd = 1;
    } while ( __HAL_TIM_GET_COUNTER(&s_TimerInstance) < 9);
        txd=0;
        count++;

    switch (state ){                                                  //State machine for fault detection
    case 0: // Ignore the first high fault, giving txd pulses to clear the fault without reading fault code
        if (count >= 26 && status == 0 ) {
            count = 0;
            state = 1;
        }
        break;

    case 1: // Fault pin is low and giving 100 fault pulses/waiting for fault pin to go High for second time
        if (count>=100 && status == 1 ) {
            count = 0;
            state = 2;
        }
        break;

    case 2: // Second time fault activated, need to read the fault code(10 pulses + 6 pulses to read the rxd +
+10 pulses to clear the fault
        if (status == 1){
            if( (rxd.read() == 1 || rxd_read == 1) && i<6) {
                arr[k] = rxd.read(); //Read RXD (fault code bit) and store in array
                k++;
                rxd_read = 1; //Flag to indicate that fault has been read
                i++;
            }
        }
        else if ( status == 0 ) { // Once fault pin becomes 0, move to state 3
            fault_read = 1;
            rxd_read = 1;
            count = 0;
            state = 3;
        }
        break;
    case 3:
        if (status == 0) {
            InterruptIn fault (PA_0); //Configure fault pin as interrupt pin
            fault.rise(&fault_init); //Enable fault interrupt
            toggle_txd_i_ticker.detach(); //Disable state machine
        }
        break;
    }
}
}

```

Figure 5: Example code of the algorithm when the Fault-pin signal goes high (Source: Maxim Integrated)

```

Int tec_count= ((CAN1->ESR) && (0x00FF0000))>>16; //Get the transmit error counter value from ESR register
Int error_code= ((CAN1->ESR) && (0X00000070))>>4; //Get the error code value from ESR register
if((tec_count >255) || (error_code !=0 ))
{
    fault_init(); //Run the fault detection algorithm as implemented in above s
ection
}

```

Figure 6: Example code of the algorithm if bit-error frame and transmitter error count are higher than 255 (Source: Maxim Integrated)

detect both overcurrent and transmission failure faults. To ensure the TXD pulse width is accurate, a timer should be used. Figure 4 shows an example code to set up the timer.

Low-level APIs are preferred to configure the timer with a 2- μ s resolution. Mbed timer provides a minimum resolution of 8 μ s. In this example, the TIM2 timer in the STM32F303K8 CAN controller was used. More description details on register settings are available in the STM32F303K8 programming manual.

In case the Fault-pin signal goes high, it can go high at any moment within a CAN frame transmission. That means the frame most likely ends after the Fault-pin signal is high, therefore reading the error code from the next fault detection cycle is more reliable. After the Fault-pin signal becomes high, the CAN peripheral pins are configured as GPIOs. The fault detection algorithm will repeatedly generate 2- μ s TXD positive pulses until the RXD signal becomes high after the rising edge of the Fault pin. It is recommended to sample the RXD signal after the falling edge of the TXD signal. After the RXD signal becomes high, five more TXD pulses are required to shift out the Fault code. Ten more TXD pulses are used to disable the fault detection. Figure 5 shows an example code of the algorithm when the Fault-pin signal goes high.

For the other two cases (bit-error frame and transmitter error count >255), the Fault-pin signal does not necessarily become high. The algorithm will configure the CAN peripheral pins as GPIOs and will repeatedly generate

2- μ s TXD positive pulses until the RXD-signal becomes high after the rising edge of the Fault-pin signal. It is recommended to sample the RXD-signal after the falling edge of the TXD-signal. After the RXD-signal becomes high, five more TXD-pulses are required to shift out the Fault code. Ten more TXD-pulses disable the fault detection. If Fault-pin signal does not go high after 110 TXD-pulses, this means no fault was detected and the fault detection mode will be exited. Figure 6 shows an example code of the algorithm.

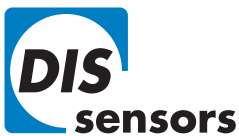
In the STM32F303K8 CAN controller, the ESR (error status register) has the transmitter error counter [TEC] bits 7:0. If this counter exceeds 255, the state machine to detect faults will get started. Also the ESR has 2 bits, LEC [2:0] (last error code) to indicate the error condition such as a bit-error frame.

◀
of

Source

Maxim Integrated (Analog Devices)

Application note: <https://www.maximintegrated.com/en/design/technical-documents/app-notes/7/7333.html>
www.maximintegrated.com



Dynamic inclinometer

AVAILABLE NOW

Stable real-time measurement,
even during rapid movement

- Gyro-compensated
- Based on MEMS technology
- Measuring range: 1 axis up to 0 - 360°, 2 axes up to $\pm 90^\circ$
- CANopen or J1939 interface (data and configuration)
- User-friendly configuration tool
- Advanced hardware and software technology
- Compact, robust housing



CAN-based drive control for a robotic manipulator

The Homer project (highly-redundant, modular robotic systems for flexible use in space and automotive manufacturing) is developing a modular robotic arm for multi-purpose and multi-mission use. A Kvaser Leaf Light HS v2 CB assures drive control and communication via CANopen, implemented using a Python CANopen API.

To minimize the need for inherently risky human interaction in servicing, maintenance, and assembly tasks in space applications, the application of robotic systems has obvious advantages. However, space robotic systems are traditionally created for specific purposes or applications, and often require a vast amount of design effort and cost, making them unfeasible for many missions.

A collaboration between the Institute of Space Systems (IRAS) at the Technical University of Braunschweig and the Institute of Structural Mechanics and Lightweight Design (SLA) at RWTH Aachen University, led to a project to develop a manipulator for use in space. The manipulator can physically adapt itself to the given location, modifying its length and degrees of freedom during operation.

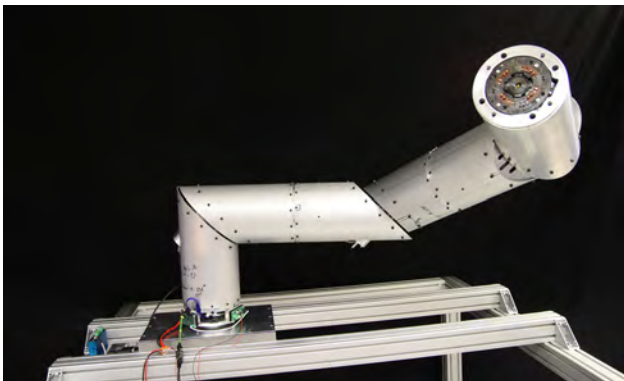


Figure 1: The manipulator can physically adapt itself to the given location, modifying its length and degrees of freedom during operation (Source: TU Braunschweig, RWTH Aachen University)

Two on-Earth demonstrators

The Homer team developed two on-Earth demonstrators; the “Little inspection and servicing arm” (Lisa) and the “Medium-sized arm for reconfiguration and grapple exercises” (Marge) to determine the operating loads, actuator and joint design, and optimize the structure. The team, comprised mostly of structural engineers, found themselves implementing CANopen to control and communicate with the drive system, without the benefit of a background in electronics.

Actronic Solutions provided valuable advice and support to the project, proposing the PCB-based (printed circuit board) CAN interfaces and supplying the servo drives. Kvaser’s Leaf Light HS v2 CB (circuit boards) were used in each module to control servo drives from Copley Controls. With

Brief facts

Project participants

[Institute of Structural Mechanics and Lightweight Design \(SLA\), RWTH Aachen University, Germany](#)

[The Institute of Space Systems \(IRAS\) of the Technical University of Braunschweig, Germany](#)

[Chair of Space Technology, Technical University of Berlin, Germany](#)

Supporting companies

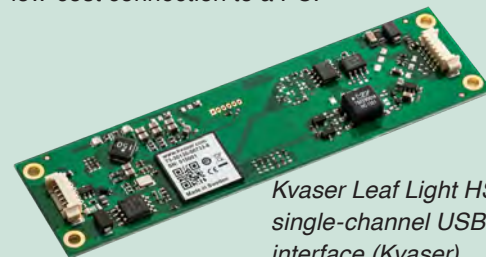
- ◆ [Actronic Solutions](#)
- ◆ [Copley Controls](#)
- ◆ [Kvaser](#)



Used CANopen drive APV-090-30 (Copley Controls)

Used CAN interface

The [Kvaser Leaf Light HS v2 CB](#) single-channel, high-speed, USB-to-CAN interfaces provided a reliable, low-cost connection to a PC.



Kvaser Leaf Light HS v2 CB single-channel USB-to-CAN interface (Kvaser)

More information

<https://iopscience.iop.org/article/10.1088/1757-899X/1226/1/012096/pdf>

‘Fully Modular Robotic Arm Architecture Utilizing Novel Multifunctional Space Interface’ by C. Zeis¹, C. A. de Alba-Padilla², K.-U. Schroeder¹, B. Grzesik³, E. Stoll³. It was published in 11TH-EASN IOP Conference Series: Materials Science and Engineering 1226 (2022) 012096, IOP Publishing doi:10.1088/1757-899X/1226/1/012096

¹ Institute of Structural Mechanics and Lightweight Design, RWTH Aachen University, Wuellnerstr. 7, 52066 Aachen, Germany

² Institute of Space Systems, TU Braunschweig, Hermann-Blenk-Str. 23, 38108 Braunschweig, Germany

³ Chair of Space Technology, TU Berlin, Marchstr. 12-14, 10587 Berlin, Germany

up to seven axes to control, this was a simple implementation requiring one CAN connection per module. High reliability was essential, hence the choice of the CAN network and CANopen as higher-layer protocol.

"We found ourselves learning all about PDOs and SDOs from scratch. However, we achieved a system that performed in the way we expected it to and at the price point required, which was why we opted for an open-source CANopen API built in Python," noted Christopher Zeis, scientific assistant for the Institute of Structural Mechanics and Lightweight Design at RWTH Aachen.

Key project elements

A key element of the project was to integrate the multi-functional iSSI interface, designed by the Iboss (intelligent building blocks for on-orbit satellite servicing and assembly) project. The iSSI combines four sub-assemblies for power, data, heat, and mechanical load transfer. It uses a short-range optical communication to transmit data between modular Iboss building blocks (Iblocks). Each iSSI sub-assembly is controlled and monitored by the Interface Control Unit, which integrates an optical CAN interface as a backup network for low-data-rate tasks.

Next project steps

The next step for the Homer team is to build a space-certified demonstrator. Funding is pending, but potential appli-

cations for the modular robotic manipulator on Earth - as in space - are plentiful. Taking the idea of minimizing the need for inherently risky human interaction in servicing, maintenance and assembly tasks, the manipulator could be used in other harsh environments such as the ocean floor or in the confined, highly-radioactive space of a nuclear reactor. The modular, highly redundant design could also bring this technology within the reach of many more projects since it increases flexibility during production, reducing tooling times and enabling swift adaptability to create a variety of products.

Zeis concluded: "Our modular concept complements rather than replaces current robots. We firmly believe the approach has a place in factories of the future and we are open to research cooperation in this domain." ◀



Author

Michael Odälv

Kvaser

mo@kvaser.com

www.kvaser.com

CANopen Miniature Pressure Transmitter CMP 8270

- Different accuracy classes i. e. 0.1 % FS typ
- Measurement of pressure and temperature
- CANopen DS301/DS404, supports CAN 2.0A/B



www.trafag.com/H72614

trafag
sensors  controls

Standards and specifications



This section provides news from standardization bodies and nonprofit associations regarding CAN-related documents. Included are also recommended practices, application notes, implementation guidelines, and technical reports.

New CiA specifications

CAN in Automation (CiA) has improved most of its specifications and technical reports regarding inclusive language. In minimum, there is a hint in the documents that CiA is committed to substitute non-inclusive terms in newly-released documents.

Recently, CiA has released the following specifications:

- ◆ CiA 457 (version 1.1.0): CANopen interface profile wireless transmission as Draft Specification (DS)
- ◆ CiA 459 series (three parts): CANopen profile for on-board weighing devices as DS
- ◆ CiA 611-1 (version 1.0.0): CAN XL higher-layer functions - Part 1: Definition of service data unit types (SDT) as Draft Specification Proposal (DSP)

Documents in DS state are part of an annual subscription option for non-members, for example the CiA 6XX series. CiA members have free-of-charge access to all documents including DSPs and WDs (Work Draft). ■

Call for experts: CiA profile for smart homes

CiA calls for members interested to develop a CiA (CANopen) profile for smart homes. Hyperpanel Lab informed CiA headquarters about the idea to use CANopen networks to control smart homes. CANopen networks are intended to be used to control heating and air-conditioning, lighting, domestic appliances, window shading, etc. CAN XL with its 10+ Mbit/s data phase bit rate and its 2048-byte payload capability could be used as a smart home backbone network. If there is sufficient interest in this topic, CiA office will organize a workshop to discuss the development of a CiA smart home profile. ■

CiA and Autosar cooperate

CiA has become an Autosar partner. It is intended to support each other in developing specifications and recommendations. The objective is to harmonize documents before they are released. Experts from both organizations are allowed to participate in meetings of the other association.



Currently, CiA grants Autosar members access to CiA documents, which are referenced in Autosar specifications. This includes also CiA documents in DSP (Draft Specification Proposal) status – generally limited to CiA members. The first document referenced by Autosar is CiA 611-1, which defines the values of the Service Data Unit Type (SDT) field in the CAN XL data link layer protocol. ■

New edition of ISO 11898-2

CiA has submitted the content of CiA 601-4 (SIC transceiver) and CiA 610-3 (SIC XL transceiver) to be included in the next edition of ISO 11898-2. This standard will specify all kind of CAN high-speed physical medium attachment technologies. The responsible ISO working report has already prepared a draft document, which is currently in DIS (Draft International Standard) ballot. National standardization bodies will vote on this document and may submit comments. The DIS ballot is open for three months. ■



CAN in the outer space

Since several years, CAN-based networks are used in satellites. In some of them, the classic CANopen application layer and communication profile (CiA 301) has been adapted. The European Space Agency (ESA) has specified an implementation guideline. ESA has implemented this with partners in an IP core. This CANopen Controller IP Core (CCIPC) provides a subset of the CANopen services. Two different variants of the IP-Core are present, CCIPC and RCCIPC; the latter one implements a subset of the CCIPC with the advantage of a consistently-reduced silicon area occupation. Both implementations feature NMT server and Heartbeat functionalities, the Default-SDO server, PDOs to be transmitted and received,

and supports PDO synchronization by means of the SYNC message.

Three years ago, ESA has issued an invitation to tender for the development of an open-source implementation of the CANopen protocol suitable for space applications. In particular, an implementation of the ECSS-E-ST-50-15C specification developed by ESA was demanded. The Lely CANopen stack was one of the candidates considered to be a promising starting point for such a development. N7 Space submitted a proposal based on this open-source stack and won the bid. Since 2021, the company (in the meantime CiA member) has been working, with support from Lely, to improve and extend the stack and to develop a comprehensive test suite. This work is carried out under a program of, and funded by ESA.

Recently, some more companies of the outer space business have joined the CiA community, e.g. Endurosat. This means, it is time for evaluating the development of further specifications for embedded CAN-based networks in satellites. This could include redundancy concepts as well as dedicated profiles. CiA office is going to organize a very first evaluation meeting for this application field. In the past, ESA has organized several CAN-in-space conferences in the Netherlands, Italy, and Norway.

Perhaps, CiA members can continue this. There are also chipmakers providing radiation-resistant CAN transceivers and micro-controllers with integrated CAN protocol controllers. ■

INTEGRATION IoT Fleet Management

for Driver

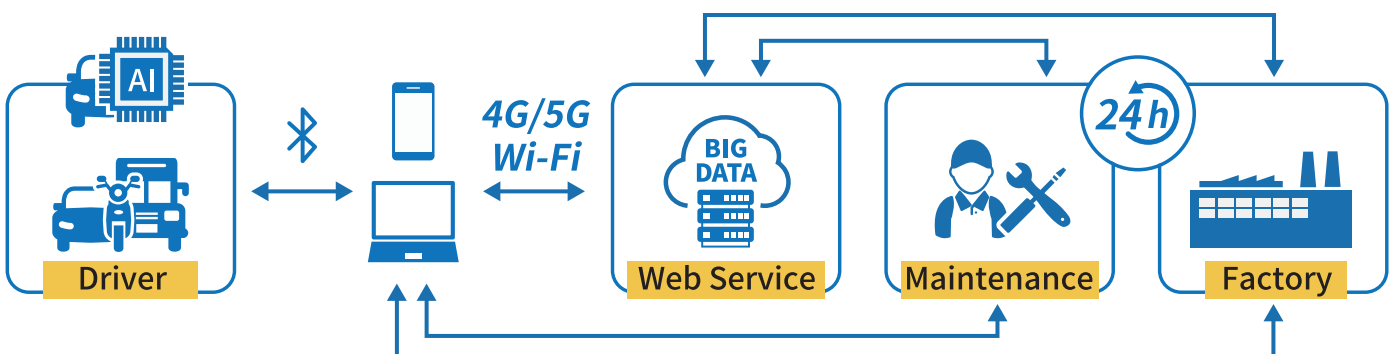
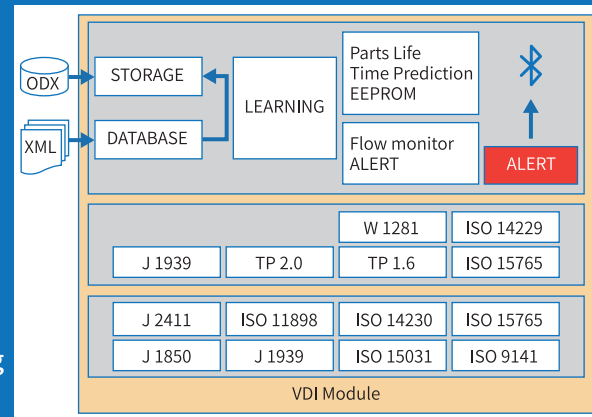
- Record data
- Upload data
- Read DTC
- Display location
- Driving behavior record
- Maintenance support

for Maintenance

- Collecting Data
- Flash programming
- Diagnostic functions
- Calibration
- Life of spare parts prediction

for Factory

- EOL Flash programming
- Diagnostic functions
- Components test
- Data analysis
- Retrofit for extended function



The new dynamic parameters of CAN SIC

The CiA 601-4 specification for CAN SIC transceivers is released and will be hand over to the updated ISO 11898-2 soon; the CiA 601-1 specification helps to understand the CAN FD high-speed transmission. The dynamic parameters are discussed in this article.

(Source: Adobe Stock)

The main difference between CAN FD transceivers and CAN SIC transceivers is the transmitter. The CAN protocol allows collision on the network. This means two or more nodes can transmit data at the same time. This is needed to support arbitration and error frames. To avoid damages on the transceiver in case of a collision, the transmitter transmits only a dominant signal on the network. The recessive signal is caused by the termination resistors only. In Figure 1 the typical behavior of a CAN FD transceiver is demonstrated.

- Phase 1: TxD is logical_1 and the transmitter output stages are high impedant and the network is recessive.
- Phase 2: TxD changes to logical_0 and the transmitter output will be switched on. The transmitter impedance changes from high to low impedance. The level on the network is dominated by the transmitting node.
- Phase 3: TxD is logical_0 and the transmitter output is switched on. The transmitter impedance is low impedant and dominates the level on the network.
- Phase 4: TxD changes to logical_1 and the transmitter output will be switched off. The transmitter becomes high impedant and the level on the network will be controlled by the termination resistors.
- Phase 5: TxD is logical_1 and the transmitter output stages are high impedant. The level on the network is controlled by the termination resistor.

reflection, the SIC transmitter changes from low impedance state in dominant to a medium impedance state during the transition dominant to recessive and for a limited time afterwards. In Figure 2 the typical behavior of a CAN SIC transceiver is demonstrated.

- Phase 1: TxD is logical_1 and the transmitter output stages are high impedant and the network is recessive. The behavior is the same like a CAN FD transceiver.
- Phase 2: TxD changes to logical_0 and the transmitter output will be switched on. The transmitter impedance changes from high to low impedance. The network is dominated by the transmitting node. The behavior is the same like a CAN FD transceiver.
- Phase 3: TxD is logical_0 and the transmitter output is switched on. The transmitter impedance is low impedant and dominates the level on the network. The behavior is the same like a CAN FD transceiver.
- Phase 4: TxD changes to logical_1 and the transmitter output stages are changing from low impedant to a medium dependant state to pull actively the differential signal ($V_{CANH} - V_{CANL}$) close to 0 V for a limited time (SIC time).
- Phase 5: TxD is logical_1 and the transmitter output stages pull for a limited time (called SIC time) the differential signal actively to 0 V with an impedance of around 100 Ω , which is the typical wire impedance of a twisted pair wire. This behavior is different to CAN FD transceiver.
- Phase 6: TxD is logical_1 and the transmitter output stages are high impedant. The level on the network is controlled by the termination resistor. Same behavior like a CAN FD transceiver in Phase 5.

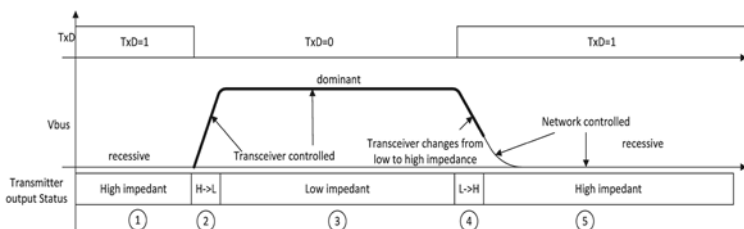


Figure 1: Typical transmitter behavior of a CAN FD transceiver (Source: Infineon)

The main difference between a CAN FD transceiver and CAN SIC transceiver is the transition dominant to recessive. The transition dominant to recessive caused ringing and reflection due to the change from low impedance to high impedance. To reduce this ringing and

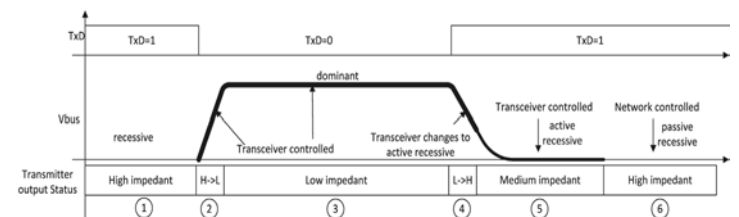


Figure 2: Typical transmitter behavior of a CAN SIC transceiver (Source: Infineon)

The maximum SIC time are 530 ns and this time might be longer than a bit time. Is the bit time shorter than the SIC time, the transmitter changes directly from SIC state into dominant state. This allows bit times which are shorter than the SIC time. The advantage is that for high bit rates one or more recessive bits in a row are supported by the SIC feature. So, ringing effects and reflection effects in a network can be longer than the bit time.

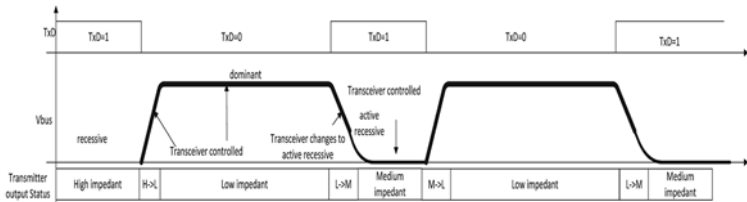


Figure 3: Typical transmitter behavior of a CAN SIC transceiver and high bit rates (Source: Infineon)

Signal improvement time

The signal improvement unit will be activated during the falling edge of the network signal (dominant to recessive transition). At which point exactly depends on the implementation and is difficult to measure. Also, for a network designer this point is not of interest. It is important to know how long the signal improvement capability is active. To make this easy to analyze and measure, the signal improvement time starts with the

rising edge of the TxD signal and ends latest after 530 ns. Figure 4 illustrates how the signal improvement time is specified.

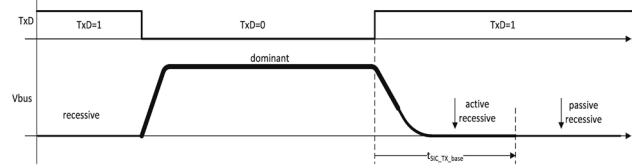


Figure 4: Signal improvement time definition (Source: Infineon)

Loop-delay symmetry

The loop-delay is the time between the TxD input signal and the RxD output signal of a transmitting transceiver. Figure 5 illustrates the transceiver loop-delay.

Figure 6 illustrates, how the loop-delay is specified. The delay of the recessive-to-dominant transition (falling edge in TxD) starts at 30 % of the TxD voltage swing and stops at 30 % of the RxD output level.

The dominant-to-recessive transition (rising edge) starts at 70 % of the TxD level and stops at 70 % of its RxD level. The loop-delay symmetry is from 70 % of a rising edge to 30 % of a falling edge of a recessive bit on RxD. The symmetry called received bit width variation is calculated according to the following formula:

$$t_{\Delta \text{Bit}}(\text{RxD}) = t_{\text{Bit}}(\text{RxD}) - t_{\text{nom}}(\text{TxD})$$



Rugged CAN Solutions For The Real World

Engineered to meet the expectations of heavy industries, the Kvaser U100 is as well suited to CAN system design as it is to CAN system maintenance and repair.

Learn more: www.kvaser.com/u100



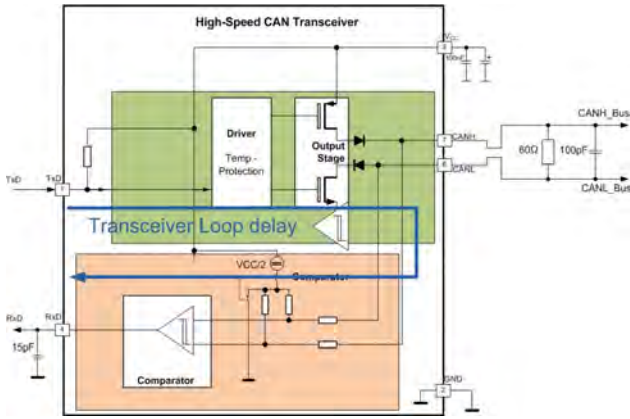


Figure 5: Transceiver loop-delay elements (Source: Infineon)

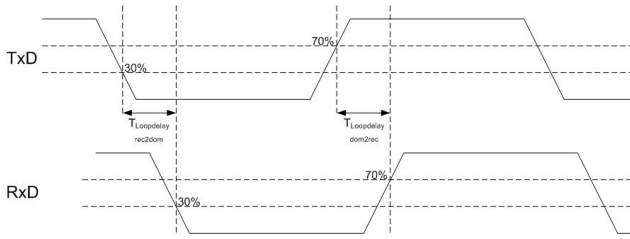


Figure 6: Transceiver loop-delay specification (Source: Infineon)

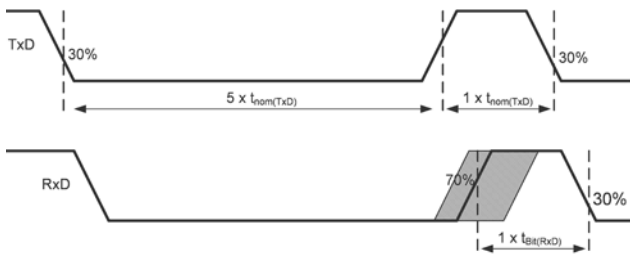


Figure 7: Transceiver loop-delay symmetry specification (Source: Infineon)

Table 1: The TxD to RxD loop-delay and loop delay symmetry parameter

Parameter set	A	B	C
$t_{\Delta Bit}(RxD)$ min	-100 ns	-80 ns	-30 ns
$t_{\Delta Bit}(RxD)$ max	+50 ns	+20 ns	+20 ns
t_{nom}	≥ 500 ns	≥ 200 ns	≥ 125 ns
$t_{loop\ delay}$	≤ 255 ns	≤ 255 ns	≤ 190 ns
Busload	60 Ω , 100 pF	60 Ω , 100 pF	60 Ω , 100 pF

Transceiver Tx delay symmetry

The Transceiver Tx delay is the time between the TxD input signal and the differential network output signal as shown in Figure 8.

The transmitter delay is the time between the rising edge of TxD signal (70%), to the falling edge of the differential signal on the network (500 mV) or the time between the falling edge of the TxD signal (30%) and the rising edge of the differential signal of the network. The symmetry is the difference between the recessive-to-dominant delay and the dominant-to-recessive delay. The symmetry is specified like for the transmitter delay. The recessive bit-length is the distance between 500 mV of the falling edge to 900 mV of the rising

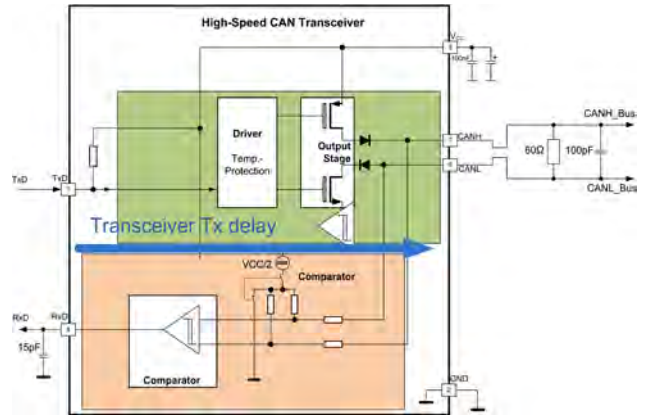


Figure 8: Transceiver Tx delay (Source: Infineon)

edge (see Figure 9). Table 2 shows the $t_{Bit}(\text{Bus})$ parameter values. The parameters are now specified as a deviation compared to the nominal bit width of the TxD signal and is valid up to the maximum specified bit rate of the device.

Table 2: Network recessive bit-width (transmitter delay symmetry)

Parameter set	A	B	C
$t_{Bit}(\text{Bus})$ min	-65 ns	-45 ns	-10 ns
$t_{Bit}(\text{Bus})$ max	+30 ns	+10 ns	+10 ns
$t_{prop}(\text{TxD-busdom})$	Not defined	Not defined	80 ns
$t_{prop}(\text{TxD-busrec})$	Not defined	Not defined	80 ns
Busload	60 Ω , 100 pF	60 Ω , 100 pF	60 Ω , 100 pF

In the current ISO 11898-2 specification the parameter Set A, are the parameters for bit rates up to 2 Mbit/s and parameter set B for bit rates up to 5 Mbit/s. The parameter set C is now for SIC transceivers.

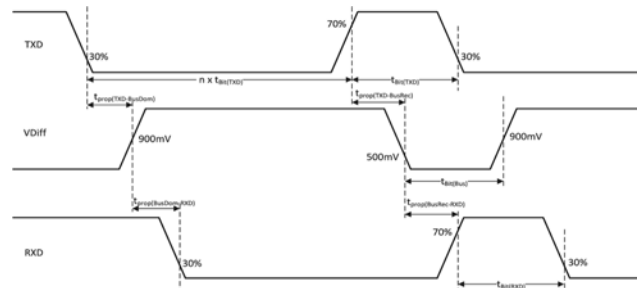


Figure 9: Specification of the transmitter and receiver delay and the bit width variation (Source: Infineon)

Transceiver Rx delay symmetry

The Transceiver Rx delay is the propagation time between the differential network signal and the RxD output signal. The definition is illustrated in Figure 9. This symmetry depends on:

- ◆ Production dispersion
- ◆ Temperature variation
- ◆ Receiver thresholds
- ◆ Supply voltage variation
- ◆ Network differential voltage (V_{diff}) slew rate

Δt_{Rec} is a calculated as follows:

$$\Delta t_{Rec} = t_{Bit}(RxD) - t_{Bit}(Bus)$$

Table 3: Tolerance of the receiver

Parameter set	A	B	C
$t_{\text{Bit(Bus) min}}$	-65 ns	-45 ns	-20 ns
$t_{\text{Bit(Bus) max}}$	+40 ns	+15 ns	+15 ns
$t_{\text{prop(busdom-RXD)}}$	Not defined	Not defined	110 ns
$t_{\text{prop(busrec-RXD)}}$	Not defined	Not defined	110 ns
Load on RxD	15 pF	15 pF	15 pF

In Table 3 the limits of the receiver delay and its symmetry for CAN FD transceiver parameter (parameter set A and B) and for SIC transceiver (parameter set C) are shown.

The new SIC feature improves existing networks and makes the communication in these networks more reliable. The more symmetric parameter allows also higher bit rates up to 8 Mbit/s. The SIC transceivers are a big step into the future of CAN and this concept is also used in the new CAN SIC XL transceiver for CAN XL communication. ◀



3 Functions
1 Device

CANnector

The flexible gateway, logging and range extender solution

- Up to 8x CAN(FD), 2x LIN, Ethernet and EtherCAT
- Easy to use, either pre-configured – unpack and get started – or flexible configurable
- Drag-&-drop configuration with included Windows-based tool
- Implementation of customized functions by adding C user code (integrated development environment)
- Enables high performance gateway solutions, logging with remote data access, OPC-UA cloud connection for alarming, and much more...

Find out more on www.ixxat.com/cannector

Author



Magnus-Maria Hell
Infineon Technologies
info@infineon.com
www.infineon.com

Securing CAN: Introduction to CryptoCAN

CryptoCAN by Canis Automotive Labs is an encryption scheme for CAN frames. It is designed to meet the requirements of in-vehicle CAN messaging. For example, publish/subscribe communications.

CAN was created in the mid-1980s to provide a robust atomic broadcast system to connect ECUs (electronic control unit) in passenger cars to replace individual signaling wires and has become a proven technology in applications as diverse as yachts and spacecraft. But CAN was never designed with security in mind – in the mid-1980s there was no notion of embedded systems being connected to the internet. Today the world is very different and there is a need to secure CAN communications because systems built with CAN are cyber-physical systems: there are actuators that move things in the real world based on the contents of CAN frames.

In mainstream computing a common way to secure communications is to use cryptography: to keep secret the contents of messages and to ensure messages have not been tampered with. This can be done for CAN systems too, but there are special requirements for CAN.

being evaluated by the United States Army Combat Capabilities Development Command (DEVCOM) Ground Vehicle Systems Center (GVSC) in the cooperative research and development Agreement “Cyber Security for Military Ground Vehicles Architectures”.

In the confidentiality integrity availability (CIA) model of communications security, CryptoCAN can provide confidentiality (i.e. keep the messages secret) and integrity (i.e. ensure messages came from a legitimate sender).

No cryptographic scheme for CAN ensures availability: attacks such as bus flooding and the bus-off attack (where a targeted device is driven offline by CAN errors) can prevent communications from taking place (just as a physical attacker can prevent communications simply by cutting the bus).



Figure 1: How CryptoCAN encodes and decodes a plaintext CAN frame (Source: Canis Automotive Labs)

Special requirements for CAN cryptographic schemes

- ◆ CAN is a broadcast network that embodies a publish-subscribe model: messages containing sensor and status information are published periodically and the sender generally doesn't know about the receivers. The cryptographic scheme must not require 1:1 communication.
- ◆ CAN is a real-time control network. The cryptographic scheme must result in messages that have bounded latencies.
- ◆ CAN frames are very small by computing standards: just 8-byte payloads. The cryptographic scheme must fit with this limited size.
- ◆ CAN systems are usually built from constrained embedded hardware. The cryptographic scheme must work on micro-controllers with limited resources.
- ◆ CAN connected devices going through a watchdog reset must return to normal operation quickly to resume control of a piece of physical hardware. The cryptographic scheme must support fast start communications.

The CryptoCAN scheme of Canis Labs is designed to meet all these requirements. CryptoCAN is currently

Basic CryptoCAN messaging

CryptoCAN takes a classical CAN frame (the plaintext frame) and converts it into a CryptoCAN message (the ciphertext message) that is sent on CAN then converted back into the original plaintext CAN frame by each receiver (Figure 1). A CryptoCAN message is 128 bit long and contains:

- ◆ The original frame payload (up to 64 bit)
- ◆ The original frame DLC (data length code, 4 bit)
- ◆ A message authentication code (MAC) of 60 bit

A MAC is a bit like a CRC (cyclic redundancy check) but much bigger and practically impossible to forge. CryptoCAN uses the standard AES-CMAC (advanced encryption standard - cipher message authentication code) algorithm to produce the MAC.

The message is encrypted using the standard AES-128 algorithm and the cipher feedback (CFB) mode. The result is a 128-bit ciphertext block. This is split into two pieces and put into two 64-bit (8 byte) CAN frames: Frame A and Frame B.

The CAN-ID (identifier) for the pair of frames is the plaintext CAN frame's ID with one bit of the ID used as the B Flag: this is 0 for Frame A and 1 for Frame B. The flag is there to ensure that the receiver can reassemble the pair of frames back into the CryptoCAN message before ▶

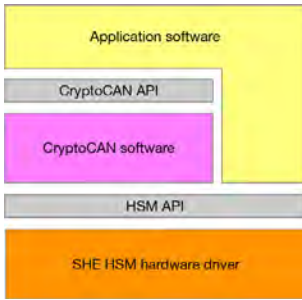


Figure 2: CryptoCAN on a micro-controller with an SHE HSM (Source: Canis Automotive Labs)

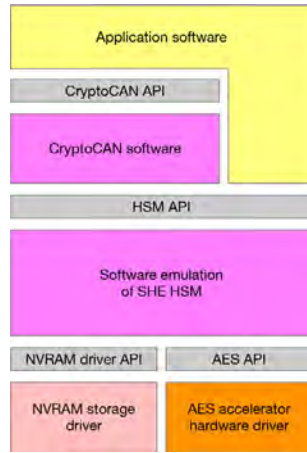


Figure 3: CryptoCAN on a micro-controller with AES accelerator hardware (Source: Canis Automotive Labs)

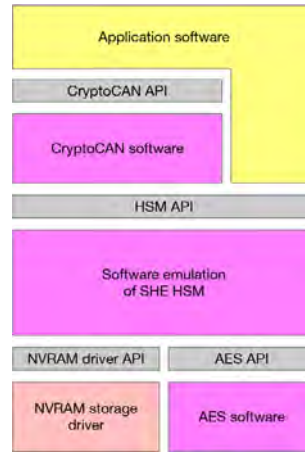


Figure 4: CryptoCAN on a micro-controller with no cryptographic hardware (Source: Canis Automotive Labs)

hardware extensions (SHE) HSM defined by the automotive industry. The SHE HSM standard specifies the AES-128 algorithm (for encrypting blocks of data) and the AES-CMAC algorithm for creating and verifying a MAC. The standard also defines how keys are managed: they are stored in secure non-volatile memory (in a dedicated area of memory that is not directly accessible by the application software), there is a defined protocol for programming them, and keys have defined permissions: they can be used for encryption/decryption or for MAC creation/verification. CryptoCAN uses the SHE HSM functions for encryption and MAC generation and verification (the keys are programmed into the HSM as part of provisioning a device).

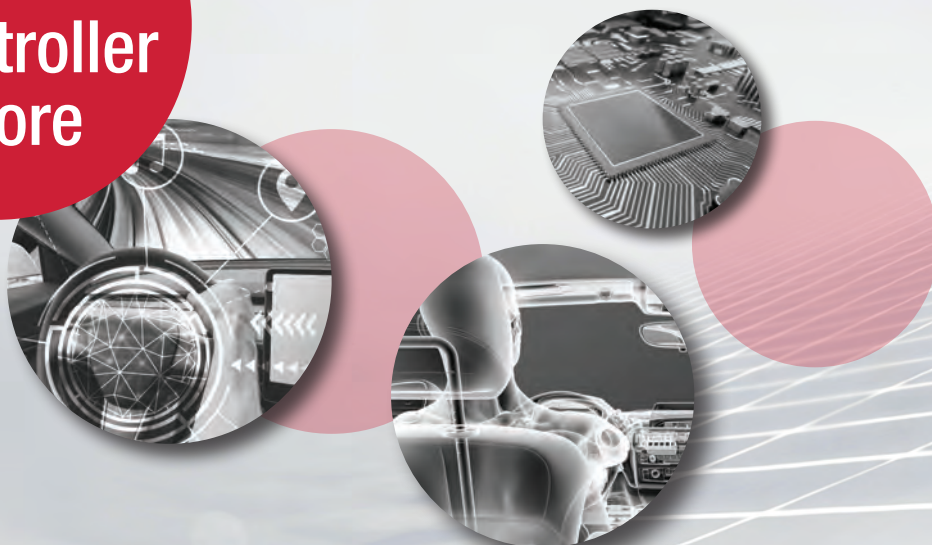
decoding. Under the CAN protocol arbitration rules, Frame A is a higher priority than Frame B and is always sent on the network ahead of Frame B. In a J1939 system the lowest bit of the priority field (bit 26) could be used. In a CANopen system, one of the address bits could be used.

CryptoCAN implementation

The cryptographic algorithms used are the ones provided by a particular hardware security module (HSM): the secure

Not all embedded micro-controllers have an SHE HSM: some have AES-128 accelerators, some have true random number generators (TRNG) and some have no cryptographic hardware. To allow CAN devices using these micro-controllers to participate in secure communications, CryptoCAN has a layered architecture (see Figures 2 to 4).

CAN Controller IP Core



CAST

Learn more at: www.cast-inc.com or email info@cast-inc.com



COMPLETE
CAN 2.0, CAN FD, CAN XL plus TTCAN AUTOSAR & SAE optimization

RELIABLE
Plugfest-verified & proven in hundreds of customer systems

FLEXIBLE
ASICs or FPGAs; Works with any Transceiver

SAFE
Certified Functional Safety ASIL D

SECURE
Available CANsec

In the first situation (Figure 2), the CryptoCAN messaging software uses SHE HSM hardware. The application accesses the HSM for key management functions (setting and updating key values).

In the second situation (Figure 3), CryptoCAN is running on a micro-controller without an HSM but with an AES-128 accelerator. In this case the CryptoCAN software includes an SHE HSM emulator that uses the AES-128 accelerator hardware via a driver API (application programming interface) and to access target-specific non-volatile memory storage (typically on-chip flash or EEPROM) to store keys.

In the third situation (Figure 4), CryptoCAN software is running on a micro-controller with-out any cryptographic hardware. There is a software emulation of an SHE HSM with a software implementation of AES-128.

A pure software implementation allows CryptoCAN to run on a wide range of CAN-connected devices. The AES-128 encrypt operation is the most compute-intensive part of CryptoCAN, and on the RP2040 micro-controller (used in the Canis Labs CANPico board) it takes approximately 13 μ s. The creation of a CryptoCAN frame requires two AES-128 encrypt operations and the decode of Frame A and Frame B, each requires one. The RP2040 micro-controller uses execute-in-place (XIP) external flash and there can be very large cache fetch delays for cache misses. Cryptographic operations must have constant execution time so the cryptographic functions in the RP2040 implementation of CryptoCAN are placed in RAM (random access memory).

CryptoCAN MAC

The CryptoCAN MAC is computed by using the AES-CMAC algorithm on 128 bits of data that both the sender and receiver are expected to know: 29 bits containing the CAN-ID (the ID with the B Flag removed, but with 1 bit set for standard/extended), 4 bits containing the plaintext CAN-frame DLC, 64 bits containing the plaintext CAN frame payload (padded if less than 8 bytes), and a 31-bit freshness value: an application-specific value representing when the frame was created (it could be a time or sequence number).

When the receiver decodes a CryptoCAN message, it computes the MAC from these same known values. If the received MAC and the computing MAC do not match exactly then the message is rejected.

The MAC will detect any tampering with a message. For example, if the payload is attached to a different CAN frame ID, then the receiver will not compute the same MAC as transmitted. Similarly, a message will be rejected if the payload is altered.

One common attack on encryption systems is a replay attack: old messages are copied and then replayed later. An attacker may not know the contents of the message but can guess from context (for example, a message may result in a door being unlocked and therefore the message contains an “unlock door” command) and they can keep copies of messages with known behaviors to replay them later. These messages are genuine (because they were created by the legitimate sender) but are not valid - because they are out-of-date. This is why CryptoCAN has a freshness

Figure 5: Interactive Micropython session on two CANpico boards (Source: Canis Automotive Labs)

value included in the MAC: after this value changes, previous messages will no longer verify.

The freshness value is controlled at the application level: it can be a shared global time kept in a real-time clock on each device, or it can be a sequence number incremented each time a message is sent. It could also be partitioned so that the upper bits reflect an operating cycle count, stored in EEPROM in each device.

One problem with obtaining the freshness value from a timer is that a message may be created at time t but be received by the receiver at time $t + L$, where L is the latency of Frame B. The freshness value at the receiver is therefore not the same as the one used to create the message, and the MAC verification would normally fail. To address this issue, CryptoCAN has an option to use the least significant 3 bits of the CAN DLC fields of Frame A and Frame B to encode the least significant 6 bits of the freshness value used to create the frames. CryptoCAN at the receiver uses these 6 bits to work out the original freshness value, determine if it is fresh, and then verifies the MAC against it.

CryptoCAN contexts

CryptoCAN creates a context for each message source: this stores data to encode and decode CryptoCAN messages, including key numbers of the encryption and MAC keys, the bit number of the B Flag, and the previous CryptoCAN message ciphertext (i.e., the payloads of Frame A and Frame B). The previous ciphertext is used by the CFB mode of encryption (a mode that allows a receiver to start receiving messages very quickly after starting or re-starting) but when a context is initialized, the previous ciphertext is unknown and set to a random value. This results in an important CryptoCAN property: the first CryptoCAN message after initialization will always be rejected. For a periodic message this is usually not a problem. But it could be a problem for a sporadic message: there may be no previous ciphertext. In this case, a simple solution is to send the message twice.

Development support

CryptoCAN software is supplied as source code with a C API. Also provided is a Micropython API to CryptoCAN in firmware for the Canis Labs CANpico hardware. This uses an RP2040 micro-controller without any cryptographic

hardware so the software emulation of a SHE HSM is included, with keys stored in external flash memory. This is of course not resilient to physical attacks (where the flash memory is de-soldered and the keys read out) but is primarily intended to be used as an evaluation kit for CryptoCAN. Figure 5 shows a simple interactive Micropython session on two CANpico boards, creating and sending encrypted CAN frames (left) and receiving and decoding them (right). The HSM on each of the CANpico boards has been pre-provisioned with the encryption and authentication keys. Note how the first CryptoCAN message is discarded.

There is further development support built into CryptoCAN: an option to disapply the encryption of CryptoCAN messages so that they are transmitted as plaintext (but still with the MAC). This helps a developer locate set-up problems (for example, failing to set the same key values at the sender and receivers) and application problems: Frame B contains the original payload and existing CAN analyzer tools can simply process the unencrypted Frame B.

Summary

CryptoCAN is an encryption scheme specifically designed for CAN. It fits the publish-subscribe paradigm common to CAN systems, where a sender is not coupled to receivers. It also supports the fast start of a receiver to participate in encrypted communication.

CryptoCAN replaces a plaintext CAN frame with a pair of ciphertext CAN frames with the same real-time properties, and where the latency of Frame B is the latency of the message, allowing existing scheduling analysis tools for CAN to continue to be used to calculate worst-case frame latencies. CryptoCAN has also been carefully designed to run efficiently on micro-controllers with no cryptographic hardware, and the extra bandwidth used by CryptoCAN is one extra CAN frame per original frame. The issue of replay attacks has been directly addressed, with support for automatically detecting and dropping replayed messages.

The CryptoCAN Micropython firmware is free to use for the Canis Labs CANpico hardware. ◀



Author

Ken Tindell
Canis Automotive Labs
ken@canislabs.com
canislabs.com

Landing soon: a more personal touch



Push to open



Showing that electric aviation is possible and beyond



(Source: Plane spotter Simeon Lüthi)

With the e-Sling project, students of the ETH Zurich (Switzerland) demonstrate that electric aviation is possible with a four-seater aircraft. Over 20 students worked on the goal of electrifying a Sling TSI as an experimental certified aircraft. CAN is on board.

The Sling TSI is a four-seater home-built aircraft from South Africa, which we have electrified in the last two years. Now we are doing 40h of test flights for the experimental certificate. And our new challenge for this year's students is to develop a fuel cell system for an aircraft to further improve the range and our field of research. The hydrogen drive train is planned to be built into a next aircraft, until then we have already started testing our system.

System architecture of the alpha project

The electric powertrain is entirely designed by us students. We developed the inverter with industry-leading silicon carbide power semiconductors, which powers the radial flux motor from our partner e+a Elektromaschinen und Antriebe. For our two battery packs we use over 2000 cylindrical battery cells with an integrated battery management system. The batteries are in the wings. Furthermore, we have a cooling system for the batteries and the motor, and of course

a control system. Here is where CAN plays an important role in our system. All devices communicate with the CAN 11-bit identifier base frame format, without any higher-level protocols such as CANopen. CAN is used in order to ensure that the main devices have all the important information they need. And additionally, that we can use sensors which also communicate with CAN. We also do not need a separate communication protocol for our graphical interface, as the data is already on the network. This is used for our HMI (human-machine interface), for the pilot, and to send the data to a background server to further analyze the data.

We use one main CAN system with nodes for the inverter, ECU (electronic control unit - with a STM32H743ZI2 Nucleo, based on a Cortex-M7), BMS (battery management system) commander, current sensors in the batteries, voltage distribution box sensor, HMI controller, and the low-voltage inverter. Furthermore, we have a second CAN system just for the charging.

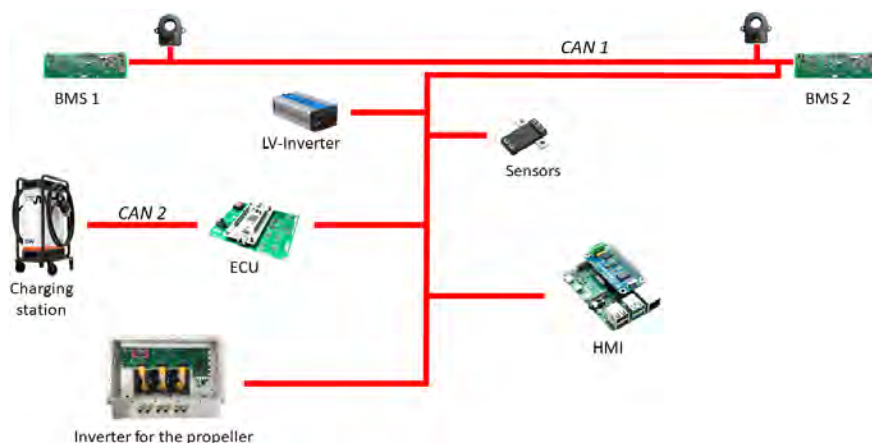


Figure 1: The CAN network in the e-Sling alpha project (Source: Cellsius)

Handling IDs and sending data

As mentioned above, we do not use a higher-level protocol. Therefore, we will explain how we handle the IDs for our nodes and how we manage to send the data. In the airplane we use the standard IDs, not the extended IDs, as we have no sensors which use the extended ID. For our data we have a large enough CAN-ID space. To know which device sends data frames, we have defined an ID space for every node. ▶



Figure 2: A range of companies and organizations supported the university to realize their project; CAN in Automation (CiA) helped to answer all CAN-related questions (Source: Cellsius)

Within the node we define which data is sent over which CAN-ID. If one measurement/data entity does not use the 8 byte of a single CAN data frame, which is mostly the case, we send several parameters either cyclic or event driven within the same ID. With this model we had the full control what we send with each frame. The downside is that it is not compatible with other systems, but as this does not matter for our purpose, we chose the path without a higher-level protocol such as CANopen.

Data recording and HMI

For the HMI in the cockpit, we use a Linux-based Raspberry Pi system to visually display all important information

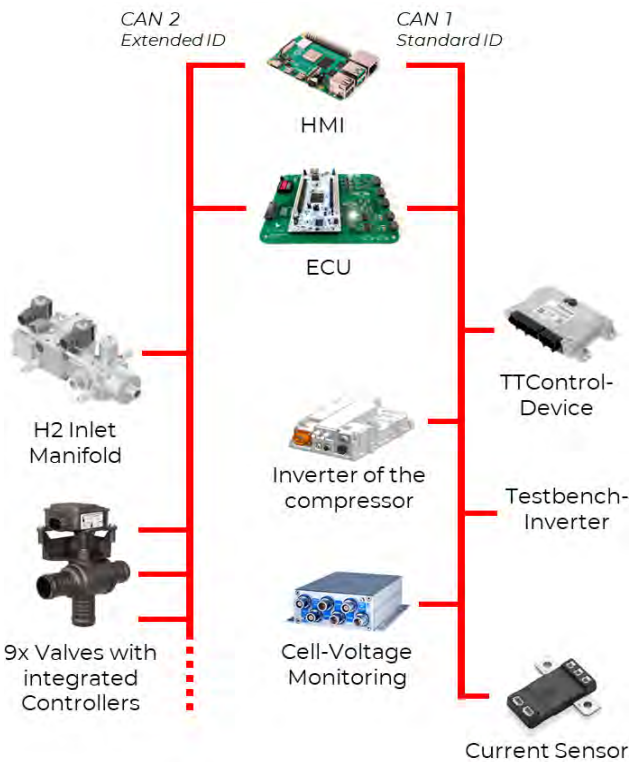


Figure 3: The CAN network in the latest project called H2 (Source: Cellsius)

to the pilot. We use the official CAN API (application programming interface) SocketCAN to read and send CAN data frames on a Linux-based system. With this approach we are very flexible to display additional data, as we can read all the data which is on the CAN network.

Furthermore, we use the Raspberry Pi to send all the data to our own server. On the server we built a surveillance tool for us developers. This graphical visualization of the data was extremely helpful for debugging purposes to support our fast development cycles. With the help of these Grafana plots we were able to detect faults early in our system, for example that one cell connection was bad. This also helped us to improve the stability of our system, especially with the electromagnetic coupling into the CAN network. It is also crucial to detect safety issues early while using the aircraft and to have an overview and feedback for the maintenance program.

CAN adjustments in the new e-Sling project H2

In our new hydrogen fuel cell project called “H2”, we also use a CAN system as the heart of our system. Here we have divided the components into a network which uses the extended ID and one that uses the standard ID. For faster development we reused our ECU and a Raspberry Pi for the server interface from the e-Sling alpha project and the methodology how we define the CAN-ID spaces. In the hydrogen system we have a lot of controllers for which we send the steering data over the CAN interface. One controller is even a MIMO (multiple input multiple output) system which we have integrated into our ECU.

This autumn we successfully demonstrated that our fuel cell works on a test bench at the PSI (Paul Scherrer Institut). The new students are now improving this core technology that it fits into the airplane. Furthermore, they are developing the rest of the hydrogen system for an airplane such as the hydrogen tank and the electric system.

Author

Timo Kleger
Cellsius – By ETH Zurich
timo.kleger@cellsius.aero
cellsius.aero/project-esling



Converting mixed sensor data to CAN (FD)

CANmod.input by CSS Electronics allows a parallel measurement of analog, digital, and pulse signals. Measured data can be bundled into CAN FD frame(s) to simplify data analysis. Solutions to log, stream, and analyze the data are provided.



Figure 1: The CANmod.input module (Source: CSS Electronics)

The [CANmod.input](#) is a sensor-to-CAN module that produces analog, digital, and pulse measurements from eight input channels. The measurement data is output via CAN or CAN FD network through a 9-pin Dsub connector. The module is able to work in standalone operation, meaning that no PC is required. The unit can be integrated in a CAN (FD) network to provide data for ECUs (electronic control unit) or CAN (FD) tools. For example, it can be used as an add-on for CSS Electronics' [CANedge](#) data loggers, which record CAN/LIN data - including from CANmod devices. It is also possible to daisy-chain multiple modules to have 16, 24, 32, and more channels.

The device can either broadcast the data onto the CAN (FD) network or provide it on-request. The available CAN (FD) channel supports 11-bit and 29-bit CAN-Identifiers. Termination can be toggled via a switch. Retransmission of frames that have lost arbitration or been disturbed by errors is supported. The 70-gram module is sized 52,5 mm x 70 mm x 24,5 mm and is IP40-protected.

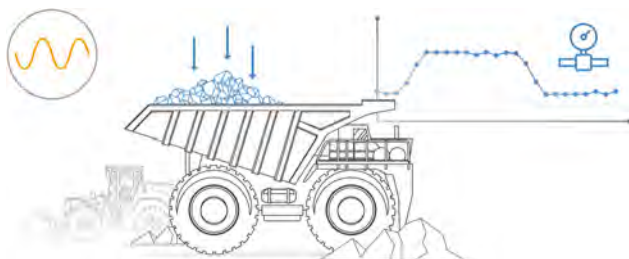


Figure 2: Analog input sensors (Source: CSS Electronics)

Common use cases

A common use case is to measure data from analog sensors. The module supports 0 V_{DC} to 10 V_{DC} analog inputs, which it can sample at 1 kHz with a configurable input range for optimal resolution/amplification. Sensors that produce analog signals can include force, pressure, current, distance, rotation, temperature, hall effect, humidity, acceleration, etc.

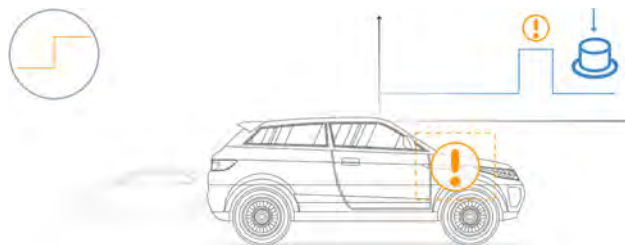


Figure 3: Digital input sensors (Source: CSS Electronics)

In parallel with performing analog measurements, the device can perform digital measurements by allowing the end user to configure the digital high/low thresholds and optional hysteresis. This enables measurement from digital input sensors such as hall effect switches, buttons, reed switches, RTD (resistance temperature detector) sensors, and more.

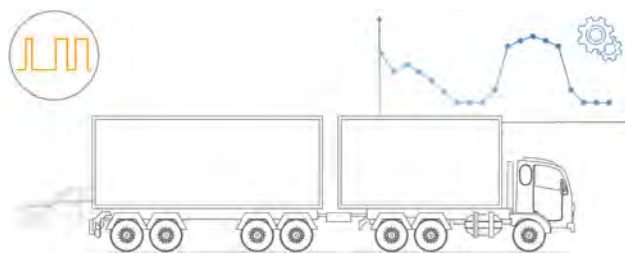


Figure 4: Pulse input sensors (Source: CSS Electronics)

Finally, the device also supports sensors with pulse-oriented outputs. These include e.g. rotational speed sensors, rotational position sensors, buttons, toggle switches (for event counting), and so on. Each pulse channel can be read at 16 kHz with configurable frequency/counter mode.

The device is able to measure the analog, digital, and pulse signals across the eight channels in parallel, which allows for mixing the mentioned sensor types. Further, the module provides a 3,3-V excitation signal for supplying the input sensors.

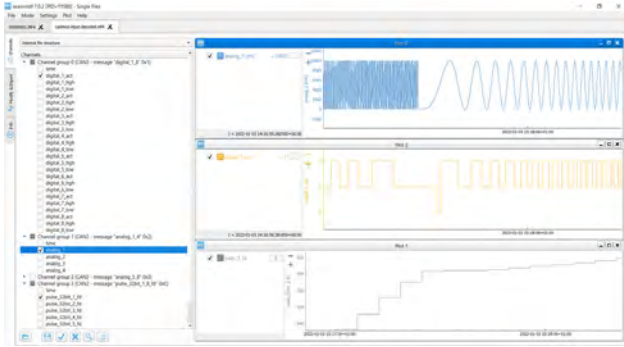


Figure 5: The device measures analog, digital and pulse signals of each input sensor in parallel (Source: CSS Electronics)

Configuration is done via the device USB port and a GUI (graphical user interface) editor. Here, it is possible to customize the output bit rate, the CAN-Identifiers, CAN frame frequencies, and further features. In particular, the CANmod.input optionally supports bundling signals into CAN FD frames. The 64-byte payload of a CAN FD message is suitable for bundling together the high-resolution signals from all eight input channels into a CAN FD frame with a single timestamp. This simplifies the data analysis.

Example use cases

The input-to-CAN module can be used in standalone mode to inject CAN frames with analog/digital/pulse sensor mea-

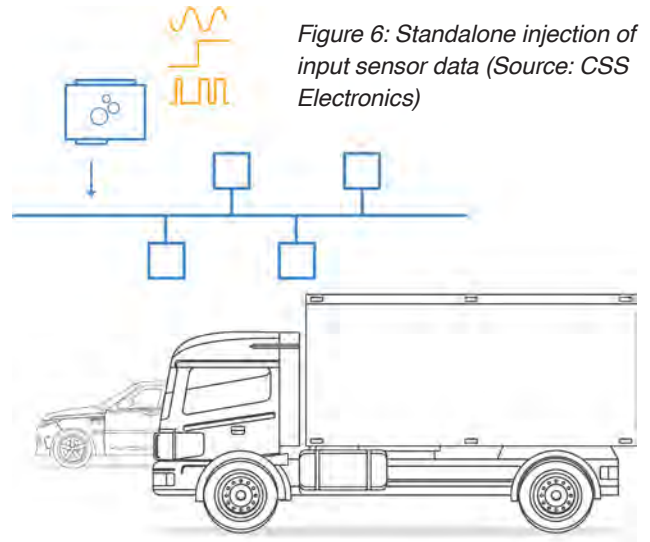


Figure 6: Standalone injection of input sensor data (Source: CSS Electronics)

surements into an in-vehicle/-machine CAN network. This allows other CAN nodes to leverage the sensor data, for example in the ECUs, cabin displays, CAN loggers, or telematics control units (TCUs).

The device is also often used as an add-on module in CAN data acquisition, meaning a CAN data logger (such as the CANedge or a CAN interface) records the CANmod.input sensor data together with e.g. in-vehicle CAN network data. Here, the input module can be daisy-chained for more channels and/or combined with e.g. the [CANmod.gps](#) or [CANmod.temp](#) products providing GNSS/IMU (global navigation satellite system/ inertial measurement unit), ▶

LIN & CAN Tools for Test and Production

LINWorks

C/C++
.NET
LabVIEW
Python

Windows/Linux

SPS/PLC

Raspberry Pi

EOL **Development** **Test**

Digital I/O PWM Analog Inputs

USB Ethernet RS 232 Digital I/O

Protocols and Services Raw, DTL, ISO-TP, UDS, etc.

Functionality Identification Configuration Traceability

SecOC **Multi-PDU** **E2E**

New Feature

CAN FD (up to 2 channels)

CAN HS (up to 2 channels)

CAN LS

LIN (up to 6 channels)

LIPOWSKY INDUSTRIE-ELEKTRONIK

SINCE 1986

www.lipowsky.com

info@lipowsky.de

+49 6151 93591-0

ISO 9001 : 2015

Distribution China: Hongke Technology Co., Ltd
Distribution USA: FEV North America Inc.

Ph: +86 400 999 3848
Ph: +1 248 293 1300

sales@hkaco.com
marketing_fev@fev.com

www.hkaco.com
www.fev.com

and temperature data. The included CANmod.input DBC file allows to decode the CAN frames to scaled engineering values during the post processing analysis.



Figure 7: Add-on module for data acquisition and telematics (Source: CSS Electronics)

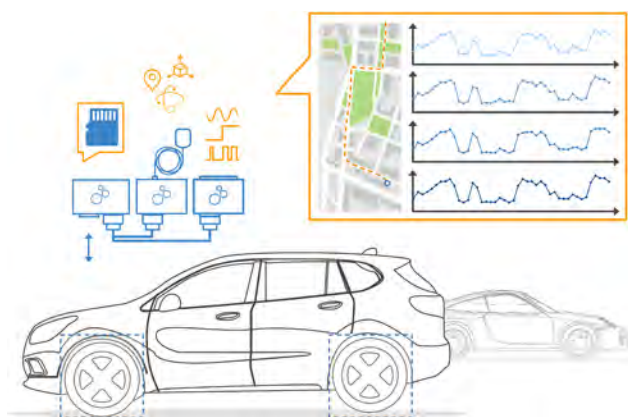


Figure 8: Data acquisition of four wheel-speed sensors (Source: CSS Electronics)

Optionally, the device can also be connected via USB to a PC to enable real-time streaming of raw or decoded CAN data into the [SavvyCAN GUI](#) tool.

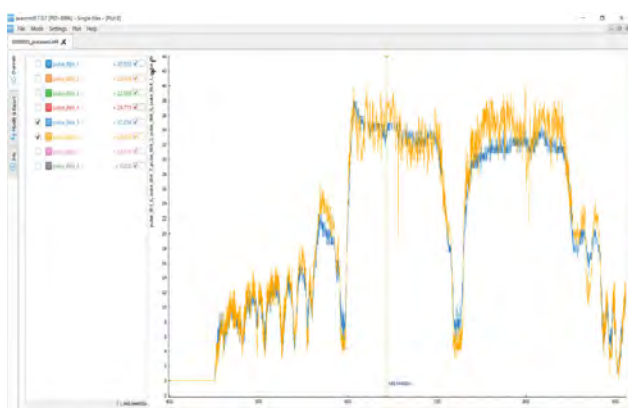


Figure 9: Analyzing the DBC-decoded signals in Asammdf GUI (Source: CSS Electronics)

Case study

ARC Vehicle (UK) is a mix of engineers, designers, and business leaders working across diverse vehicle projects and beyond - including high-performance and luxurious vehicles.

In a [case study](#), they utilize the CANedge1, CANmod.gps, and CANmod.input to collect data from four wheel-speed sensors in a test vehicle. This serves as an example of how the input-to-CAN module can be used for measuring pulse frequency inputs. The team used the open-source [Asammdf GUI](#) to analyze the input sensor data in combination with time-synchronized GPS (global positioning system) plots. This enabled a rapid solution for their monitoring needs. ◀

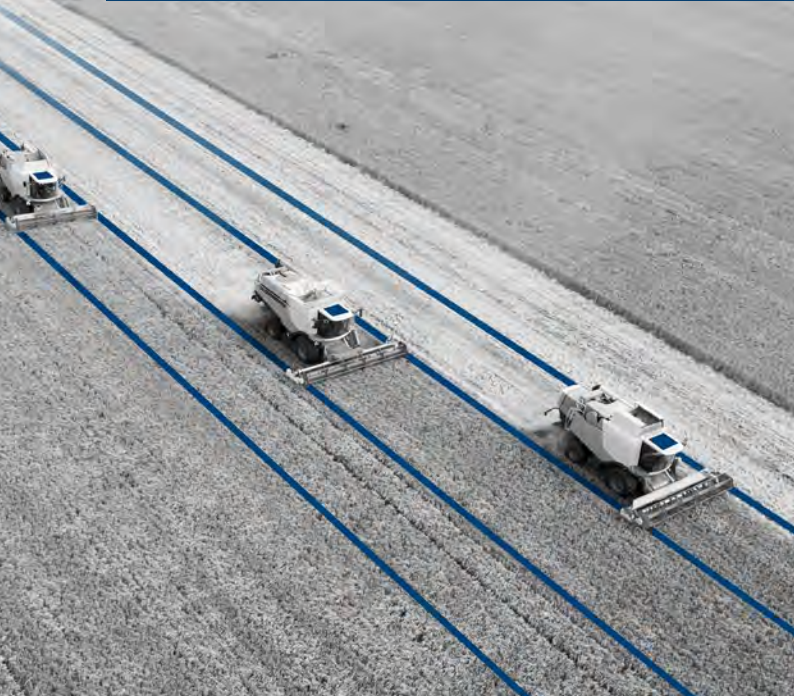
Author



Martin Falch
 CSS Electronics
contact@csselectronics.com
www.csselectronics.com

High-end Connectivity and Data Management

Telematics and Cloud Systems for IoT and Service 4.0



Continuous digitization for smart vehicles

Modular on-board units – from cost-saving entry telematics up to high-end modules. Including updates-over-the-air, embedded diagnostic functionality and up to 4x CAN channels (CAN FD ready).

Sontheim IoT Device Manager and IoT Analytics Manager – for a highly secure, comfortable and individual visualization as well as management of your data and fleet.

COMhawk® xt – Telematic ECU Series



Up to 4x CAN acc. to ISO 11898



LTE, WLAN, Bluetooth, LAN



Multi-protocol support (J1939, J2534, UDS, KWP, ...)



Positioning (GNSS)



Embedded diagnostic functionality



Integrated update-over-the-air functionality



Mobile configuration, service, and diagnostic access to CAN systems

The solution from IoTize brings secure mobile access to CAN systems from phones and tablets. CAN data can be visualized on and created from Android and iOS platforms using an app builder.

When it comes to access from mobile platforms like phones and tablets to CAN systems, then so far solutions often required customize coding to exchange data with and visualize on Android or iOS platforms. The latest solution by IoTize simplifies the process and brings easy and secure mobile access to CAN systems based on a combination of NFC, Bluetooth, and WLAN communication. The Tapioca gateway illustrated in Figure 1 has just 5 connections (Power, CAN, shield) and measures 8 cm x 5 cm x 2 cm. It is equipped with multiple wireless interfaces: NFC, Bluetooth, and WLAN.

The Tapioca hardware interface is only one of the devices of the entire solution provided by IoTize. The configuration of the interface is handled in the IoTize Studio software provided at no extra charge. It can import the commonly used “candb” format which is used to define signals in CAN frames. The imported data immediately allows selecting signals from the CAN system by point and click. The integrated app builder allows connecting graphical input and output elements to the selected signals. A locally generated HTML page shows the layout as it will later be presented in the app. The screenshot in Figure 2 shows the “Resource View” of the software with the imported signals.

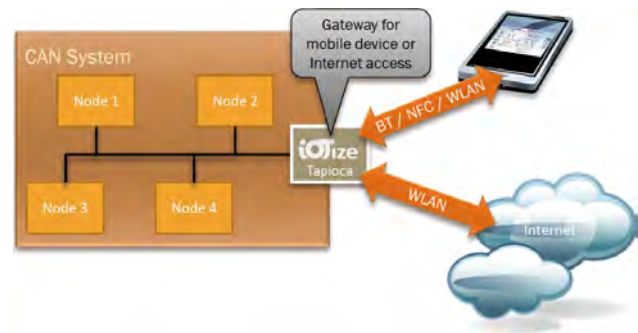


Figure 1: IoTize Tapioca communication interfaces (Source: Emsa)

From CAN data base file to your custom app

Figure 3 illustrates the workflow involved to create a mobile app to access your CAN system. First you need a candb/dbc file with the CAN signals in your system. Import this to IoTize Studio. In IoTize Studio, configure your mobile app. Which signals are used and displayed where and how? All imported signals can be selected and associated with a graphical element such as gauge, slider, chart, pie, and others. Further configuration includes which of the communication channels (NFC, BT) are used. Once you com-

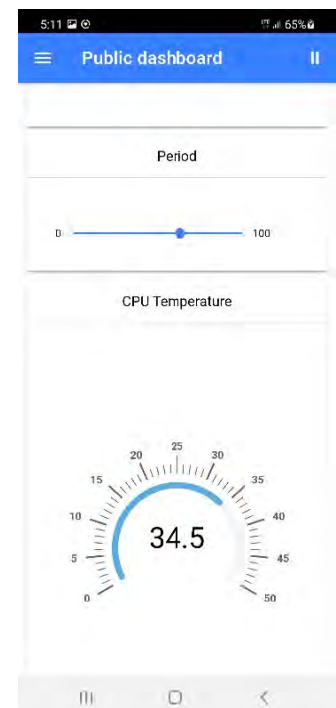
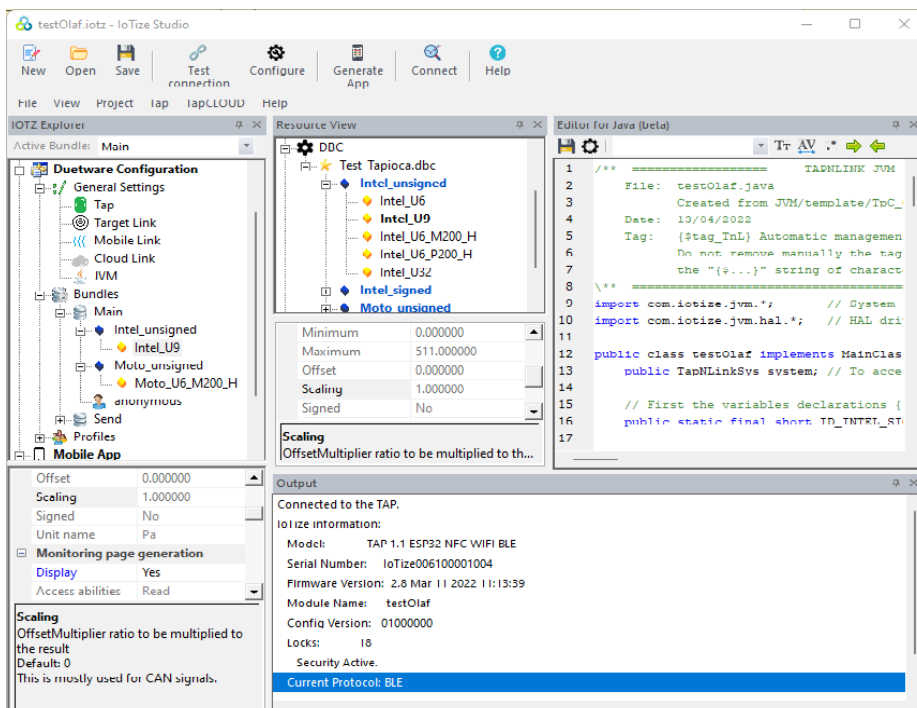


Figure 2: IoTize Studio configuration software with imported signals and app preview (Source: Emsa)



Figure 3: IoTize configuration workflow (Source: Emsa)



Figure 4: IoTize login and authorization (Source: Emsa)

pleted the configuration, it can be transferred to the IoTize server building to your custom app that you need to download and install on your mobile device. The same configuration is also transferred to the Tapioca device(s) that later provide(s) the access to the CAN system.

After the app is installed and access has been granted (e.g. through NFC authorization), the mobile app can connect to the gateway and display and set the configured signals.

Access management

Access authorization is managed by creating user profiles in IoTize Studio. The user profiles are loaded into the Tapioca devices as part of the configuration data. Only mobile apps that authenticate themselves with the proper user profile credentials get access to the gateway and ultimately to the CAN system. As shown in Figure 4, no servers are required for the login process.

The configuration loaded into the Tapioca interface may contain a custom Java program. On this level, Java has full access to all signals and also to the various wireless interfaces. This is illustrated in Figure 5. Using MQTT services, data signals can be transferred securely via an MQTT broker, allowing the mobile app to also access the CAN system remotely. Another potential use case could be to use the locally-connected mobile device as an Internet

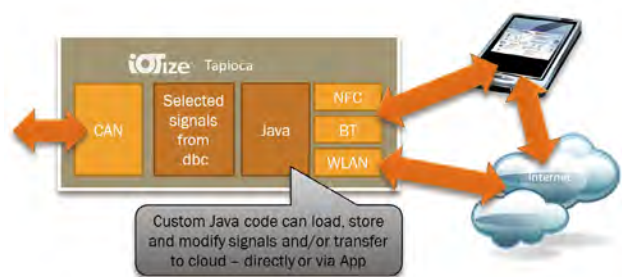


Figure 5: Advanced automated signal handling with Java (Source: Emsa)

gateway and transfer signals to a cloud using MQTT services. If the Tapioca interface is directly connected to a hotspot, such transfers can also happen directly, without the mobile device being present.

IoTize and Emsa

Emsa supported IoTize in CAN-related matters as a consultant and helped them with a CAN and potentially CANopen interface for their product line. Emsa accompanied them to define and test the first prototypes. For the IoTize CAN hardware, Emsa will also operate as a distributor. ◀



Author

Olaf Pfeiffer
 Emsa (Embedded Systems Academy)
info@esacademy.com
www.esacademy.de



CiA email-services

If you have trouble viewing this email, you can [display it in your browser](#).

CAN Info Mail

December 2022

This monthly email is for all CAN fellows

A word from the CiA Managing Director

The year of CiA's 30th anniversary is coming to its end. In June, we had a wonderful two-day birthday event. I met in person old and new CiA fellows. After two years with just online meetings because of the Covid-19 pandemic, this was very pleasing. In this year, CiA participated again in some trade shows (Innotrans, Bauma, and SPS). The success was not overwhelming, but with the still existing travel restrictions we were rather satisfied: "it was not that bad" (British understatement).



(Source: Adobe Stock)

December issue of CAN Newsletter magazine

The fourth and last issue of the free-of-charge CAN Newsletter magazine 2022 has been released. It continues with the "history and trends" series featuring 'CAN in healthcare' and 'CAN in non-automotive applications'. CiA's Holger Zeitwanger gives an outlook on 'CiA's future and CAN XL'. A range of applications reports are also provided and include: 'CAN-based drive control for a robotic manipulator', 'Showing that electric aviation is possible and beyond', 'Converting mixed sensor data to CAN (FD)'. Topics such as 'The new dynamic parameters of CAN SIC' and 'CAN transceiver fault detection with algorithm' are also covered. Additionally, 'Securing CAN: Introduction to CryptoCAN' and 'Mobile configuration, service, and diagnostic access to CAN systems' comprise the magazine. Brief news are provided in the section standards and specifications. The articles can be downloaded individually or you can download the entire magazine.



If you have trouble viewing this email, you can [display it in your browser](#).

CiA Member News

This monthly email is for CiA members only.

November 2022


A word from the CiA Managing Director

CAN SIC (signal improvement capability) transceivers as specified in [CiA 601-4](#) are gaining acceptance. The benefit is obvious: The system designer has more flexibility regarding network topologies including the length of not terminated stubs, and the network overall length at a given arbitration bit rate. But there is also a trade-off related to the CAN SIC concept: a limitation of the arbitration bit rate as explained in Annex A.1 of CiA 601-4. Rational is, that multiple nodes execute the "Active Recessive Drive" to the network lines while another node need to read dominant state of the winning node for the arbitration process, all "SIC" nodes driving "Active Recessive" need to have finished their SIC time before the sample point is reached for all nodes all over the network. This limits either the arbitration bit rate respectively the maximum node distance up to 255 ns. These arbitration bit rate limitations do not apply, if all nodes of the network change to recessive at the same bit position.



Bauma 2022 review


CiA has been a part of the Bauma tradeshow for construction, building materials, and mining machinery industries in Munich (October 24 to 30, 2022). We reported about the CAN-related novelties from the fair in our [CAN Newsletter Online](#).



CiA home game: SPS 2022

CAN in Automation has also successfully participated at the SPS exhibition, hosted in its home-town Nuremberg from November 08 to 10, 2022. About 70 product panels from 31 CiA member companies have attracted visitors to the CiA stand.

The CAN Newsletter also [published several articles](#) on products and developments found regarding CAN.



In case you like to receive an monthly update on:

- ◆ Latest trends in CAN-based networking
- ◆ In-depth articles about CAN-based solutions
- ◆ CiA events
- ◆ CAN products
- ◆ CiA community
- ◆ Participation opportunities in CiA activities

Register for the CiA email-services, by writing an email to: mail@can-cia.org
www.can-cia.org