CANopen host controllers - today and tomorrow

Reiner Zitzmann (CAN in Automation)

Host controllers in embedded networks are responsible for a comprehensive network configuration, during start-up and operation. They manage the network, with special regard to the current status of the application. They take care for the availability, the energy consumption, the safety, and the cybersecurity of the application. Additionally, they may host a gateway to web-based applications.

This paper provides an overview about the tasks of embedded host controllers and the harmonized solutions, provided in CiA's CANopen specifications. Additionally, the update of these specifications with regard to CANopen FD and today's requirements on future embedded network management are discussed.

I. Introduction

Host controllers in embedded networks are responsible for a comprehensive network configuration, during start-up and operation. They manage the network, with special regard to the current status of the application. They take care for the availability, the energy consumption, the safety and the cybersecurity of the application. Finally, they may provide a gateway to web-based applications. For CANopen Classic (CANopen CC), CiA has already developed a lot of CiA specifications suitable for host controllers and the aforementioned tasks. For CANopen FD, CiA technical working groups are still at the beginning in updating these specifications. CANopen FD does not just benefit from an improved data link layer. It offers also comprehensive services for system configuration and maintenance, which is particularly very advantageous for host controllers. Furthermore, the update of services for host controllers considers that today's microcontrollers are more powerful and provide much more resources than in the past. Therefore, new or updated CANopen FD services for host controllers can encapsulate more complex tasks and can simplify the job of an host controller.

Embedded host controllers play a vital role in facilitating communication and management within CAN-based systems, particularly those utilizing the CANopen higher-layer protocol. This paper delves into the current capabilities of embedded host controllers in CANopen-based applications and anticipates their future evolution to address emerging trends and challenges in this domain.

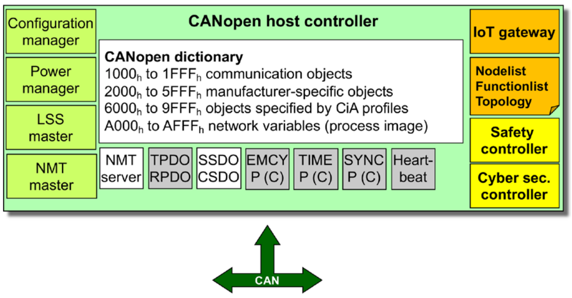
Configuration manager
Power manager
LSS master
NMT master
CANopen host controller
CANopen dictionary
1000_h to $1FFF_h$ communication objects
2000_h to $5FFF_h$ manufacturer-specific objects
6000_h to $9FFF_h$ objects specified by CiA profiles
$A000_h$ to $AFFF_h$ network variables (process image)
NMT server
TPDO RPDO
SSDO CSDO
EMCY P (C)
TIME P (C)
SYNC P (C)
Heart-beat
IoT gateway
Nodelist Functionlist Topology
Safety controller
Cyber sec. controller
CAN

Figure 1: Functions and data, residing at a CANopen host controller

II. Network start-up, configuration and management

A. General

A CANopen manager device serves as a central control unit in a CANopen network, responsible for orchestrating communication among various devices within the network.

Its primary responsibilities include network initialization, node configuration, heartbeat monitoring, and error detection and handling. By managing network parameters such as node IDs, communication bitrates, message linking and routing, the manager device establishes a robust and efficient communication infrastructure within the network.

One of the defining features of a CANopen manager device is its ability to dynamically configure and manage network topologies. Through the use of network management protocols, layer setting services or the configuration services such as (U)SDOs, the manager device can add or remove CANopen devices, assign device roles, and modify communication parameters on-the-fly. This flexibility enables seamless integration of new devices into the network and facilitates scalability to accommodate evolving system requirements.

Furthermore, the manager device plays a crucial role in ensuring the integrity and reliability of data transmission within the network. By monitoring the heartbeat messages exchanged between nodes, the manager device can detect communication failures or device malfunctions in real-time, triggering appropriate error handling mechanisms to maintain network integrity. Additionally, the manager device may implement advanced features such as synchronization and time-stamping to facilitate coordinated operation among networked devices.

For basic tasks of embedded host controllers, CiA has already specified since years, the CANopen additional application layer functions CiA 302. The specification provides services for network management, configuration and program download, network variables and process image management, dynamic SDO management, network redundancy, and multi-level networking.

B. CANopen host controllers a la CiA 302

According to CiA 302, CANopen host controllers are devices that enhance the basic CANopen functionality, by a CANopen manager functionality. Thus, they shall follow the CANopen specification CiA 301 that clarifies that CANopen devices have an Object Dictionary and a NMT state machine. This differentiates these kinds of devices from CANopen configuration tools and allows tools or other network participants to access the CANopen host controllers remotely by SDO, or to initiate NMT state changes.

The active CANopen NMT master resides at a CANopen manager device. The active CANopen NMT master manages all the other CANopen devices in the network, the so-called NMT servers. For the active management, the NMT master uses the NMT protocol.

To increase the availability of a CANopen application, CANopen has introduced a “multi-master” support. There may be several master-capable devices within a CANopen network. These devices monitor each other. According to an application-specific priority assignment, the highest prior master, acts as active CANopen NMT master resp. CANopen manager. All the other devices act as “simple” CANopen devices and monitor the CANopen manager. In case of a severe error situation, any of the other master-capable CANopen devices can overtake the tasks of the CANopen manager. The method of overtaking can be based on the flying master protocol, as specified in CiA 302.

Furthermore, at the CANopen manager there resides the “configuration manager”. The Configuration manager provides mechanisms for the configuration of CANopen devices in a CANopen network during boot-up and system runtime. Thus, CANopen managers support data bases in their object dictionary that enable them to verify the correct origin as well as the correct configuration of CANopen devices, attached to a CANopen network. CiA 302-2 specifies the list of CANopen devices, expected for a proper operation of the application. Furthermore, the expected device type, vendor-ID, product code and serial number, as well as the configuration may be provided. In case the CANopen manager identifies that a CANopen device does not follow the expected configuration, the CANopen manager may be enabled to modify the device’s configuration, according the current needs. As CANopen manager typically claim the devices’ first SDO server channel, they have access to any CANopen device’s object dictionary and can modify any configuration setting within a device individually.

Alternatively, they could have the option to modify the settings of the device by updating the device firmware entirely. According to CiA 302-2 and -3, CANopen managers can administer within their CANopen object dictionary the intended device configurations, in form of so-called Concise Device Configuration Files (CDCFs). Additionally, they may have access to a data base with CANopen device firmwares that are downloaded to the related CANopen device. CiA 302-3 provides the entire toolbox to doublecheck the correctness of a device firmware, to switch a CANopen device to bootloader mode and to update a device's firmware, including the intended configuration(s).

Furthermore, the CANopen manager may have the task to recognize attached CANopen devices, and to enable them to start into the CANopen network, by assigning them the appropriate CANopen node-ID. For this task, CANopen complements CANopen managers by means of the Layer setting services (LSS). These services are specified in CiA 305. CANopen managers, hosting the LSS master functionality can ask whether there are still devices in the network that demand a CANopen node-ID (so-called unconfigured CANopen devices).

By means of the LSS services, the LSS master can identify these devices and can switch them individually to a LSS configuration state, where a unique node-ID value can be assigned to the device. Thus, one device after the other can be identified and started. To accelerate the time for scanning and identifying "unknown" devices in the CANopen network during the overall lifetime of a given system, the CANopen manager may dynamically store identification parameters such as device type, vendor-ID, product code, revision number, serial number, etc. in the aforementioned parameters, introduced for configuration managers.

With special regard to busload management, the CANopen manager may also host the producer functionality for the SYNC. The SYNC service, introduced in CiA 301, is transferred cyclically, usually with the intend to control the busload. The transmission of the process data of the CANopen devices may be coupled to the occurrence of the SYNC messages in such a way that an almost equalized busload is achieved, burst conditions are avoided, and a certain degree of determinism is introduced in the event-driven communication system CAN.

Typically, CANopen managers provide the process image of the application. Thus, they may use the network variables (specified in CiA 302-4). To keep the process image up to date, and to refresh also diagnostic and configuration data that is not frequently transmitted via the CAN, CiA 314 provides harmonized function blocks to gain access to that data, in particular from point of view of a CANopen manager device. Among others there are function blocks available for the SDO access to any CANopen device in the network.

Modern CANopen-based applications are typically not based on a single CAN. Several CAN segments are combined by means of routers and gateways to an entire system. The remote access specified in CiA 302-7 allows CANopen managers, located at the systems top-level, to access deeply embedded devices, in cascaded CANopen-based applications. The keep the overview over such kind of architectures, assembled of several CANopen networks, CANopen managers can use the "node list", as provided in CiA 306-3. Per supported CANopen network, the file "nodelist.cpj" may identify the specific network, the number of attached devices, as well as their status, whether a dedicated device is available or not. Per device, the file allows the description of the device's "name", configuration file name, as well as the name and the path to the electronic data sheet. (CAN FD).

III. Extended host controller functionalities, already introduced for CANopen CC

A. CANopen host controllers with IoT gateway functionality

The information given in the node list is not only useful for the CANopen manager directly. The CANopen manager may also host the gateway to IoT applications and web-based services. Thus the CANopen manager may have the task to display available embedded resources to the “IoT”.

CANopen managers can follow the harmonized network access services, as specified in CiA 309 to offer access to the configuration, process, and diagnostic data, generated in the embedded network. A host controller implementing CiA 309 gateway functionality may support one or more of the following classes:

- class 0: The gateway is a device, acting as simple device (NMT server functionality) within the CANopen network. The device is intended to retrieve the data from the CANopen network;
- class 1: The gateway is a device, acting as simple device (NMT server functionality) within the CANopen network. The device provides SDO client functionality;
- class 2: The gateway is extended to support the functionality of a class 1 device with SDO requesting device (SRD) functionality;
- class 3: The gateway is a device acting as the CANopen manager (as defined in CiA 302-1) within the CANopen network;
- class 4: The gateway is extending class 3 by supporting also the LSS master (CiA 305).

Thus, a CANopen host controller, which is typically the host of the current process image of an application, can provide web-based applications, insights to the embedded system, scalable from a simple monitoring function up to full configuration and control access.

Latest amendments to CiA 309 allow a comprehensive user and resource management. The resource management system within the CiA 309 gateway application is fundamental for ensuring efficient utilization of resources allocated to registered users. The management procedures are tailored to the specific implementation of the gateway, adapting to its capabilities and requirements.

One critical aspect of resource management involves monitoring the number of users concurrently accessing the gateway. Depending on the gateway‘s resources, a limited number of users can be supported in parallel. Information regarding the maximum number of users and the current user count can be obtained using the user_info command, allowing administrators to monitor and regulate user access effectively.

Access to remote nodes is meticulously controlled to prevent conflicts and ensure exclusive user rights. Users must request access to remote nodes through the register_remote_node command, thereby preventing multiple users from operating on the same remote net or node simultaneously. The gateway validates and grants exclusive access rights to net or node combinations, ensuring smooth and conflict-free operation. Additionally, registered remote nodes can be easily unregistered using the unregister command, facilitating efficient resource allocation and user management. Automatic “unregistration” resp. logout of all nodes associated with a user occurs upon the user‘s logout or disconnection, ensuring no resource conflicts persist.

Configurations play a vital role in tailoring the gateway‘s functionality to suit user requirements. The gateway maintains separate configuration sets for single-user and multiple-user modes. Super users have the privilege to store configurations in non-volatile memory, which are then automatically applied to new users after executing the set_user_level command. These configurations are selectively processed based on the user‘s unlocked command subset, with events related to remote nodes generated only upon the registration of corresponding nodes by the user.

Memory allocation is crucial for ensuring smooth operation of the gateway. Volatile memory is reserved and assigned to each connected user, with the allocated amount being manufacturer-specific but sufficient for SDO commands. Users have the flexibility to allocate additional volatile memory for specific operations, up to the value returned by the system for the next operation, thereby accommodating varying memory demands as needed.

User management and authentication are streamlined through the login system, where users set their user level to unlock corresponding functionalities. Levels include Simple user, Standard user, and Super user, each granting progressively more privileges within the system.

Overall, the comprehensive resource management system outlined within the CiA 309 gateway, located in the CANopen host controller, ensures efficient allocation and utilization of resources, seamless user access, and streamlined administration, thereby enhancing the overall performance and reliability of the gateway application.

Additionally, CANopen host controllers can offer to the cloud the ability to use “network discovery services”. The information gained by these services allows to explore from an external point of view, which devices and functionalities are supported in an embedded CANopen-based communication system.

B. *CANopen host controllers with energy consumption control*

Taking care about the energy consumption of an application gets more and more important, as the resources of our planet are limited and energy gets more and more expensive. In particular, energy management is important in mobile applications, operating on a limited amount of energy, buffered in the application’s battery system. Typically, if supported, the power management resp. the management of the energy consumption resides at the CANopen host controller. CiA specifications simplify the tasks of host controllers in this regard as well.

CANopen devices that support CiA 302-9, already support energy saving levels. These energy saving levels are configurable and controllable, from point of view of a CANopen host controller. Also, the acceptable maximum power consumption of a single CANopen device is adjustable via harmonized CANopen object dictionary entries. To monitor the current energy consumption of a connected CANopen device, the CANopen host controller may use the functionality provided in CiA 458, the CANopen device profile for energy measurements.

In particular in mobile application, entire CANopen networks or network segments may be turned into a low power consumption mode (also often called standby or sleep mode). The power management functionality, residing at the CANopen host controller may use the services provided by CiA 320, to turn devices into such low power consumption modes and to initiate the return of “sleeping” devices, to “normal device operation”. CiA 320 offers CANopen host controllers to prepare the switching to the “sleep mode”. Simple CANopen devices can finalize in a dedicated “prepare sleep state” their current tasks so that a defined decent into the low power consumption mode is possible. In case there are long-lasting, urgent tasks still to be processed, any device can indicate to the host controller, that a decent to low power consumption mode is not possible yet, and needs to be postponed. This is done by issuing a sleep mode objection. In centralized systems, where everything is controlled by the central CANopen host controller, “sleeping devices” are woken up by the host controller. The host controller may leave the „Sleep state“ whenever a local or remote wake-up reason is detected. If the host controller is leaving the „sleep state“, it requests the “wake-up” service, which consist of consecutive wake-up requests, periodically transmitted, till one or multiple wake-up confirmations are detected. A wake-up confirmation can be the detection of CAN data frames. Basically, the “wake-up” service is intended to activate the CAN transceivers and corresponding hardware of the networked CANopen devices.

IV. CANopen FD host controllers

A. *CANopen host controllers benefit from powerful CANopen FD services*

CANopen FD, released as CiA 1301, version 1.0, in 2017, provides many powerful tools for CANopen host controllers.

The lengthened PDO, capable to carry up to 64 byte of application data, allow CANopen host controllers to provide more comprehensive and more detailed process images. Although it is worthwhile to mention that therefore the demands on memory and processing power at the CANopen host controller is increased. Additionally, the more detailed diagnostic data allows host controllers to evaluate error scenarios and to start more fine-tuned counter measures.

The CANopen FD USDO (Universal Service Data Object) extends the capabilities of CANopen CC, catering to the evolving needs of modern industrial automation and embedded systems by delivering improved performance, scalability, and interoperability. The CANopen FD USDO offers the full function-coverage of the CANopen CC SDO (Service Data Object)., but extends it with additional capabilities. A significant advancement is that it allows for the transmission of arbitrary data sizes in addition to the point-to-point, also multi-, and broadcast communication.

The ability to dynamically establish communication relationships during system runtime increases the performance further. This means that connections can be established to multiple or all network participants simultaneously. Unlike the traditional SDO protocol, these connections no longer need to be handled sequentially. The highly flexible USDO allows for multiple transactions to occur concurrently to arbitrary or even the same devices.

As a result, the data throughput is significantly increased, far surpassing the benefits of a increase in communication speed. These features make CANopen FD USDO a powerful and flexible solution for modern CANopen FD host controllers in embedded systems.

CANopen host controllers can reduce start-up time by double-checking with just one broadcast USDO, whether the intended devices are in the network, whether they are of the right origin and whether they are configured in the same way. By managing groups of the very same devices, CANopen managers can configure all identical devices, fulfilling identical tasks, by exactly one USDO write access in parallel. Also, the control of systems, modified during system runtime is no challenge anymore. Due to the dynamic USDO cross communication capability, the host controller can gain access to any newly added device and can tread the device according to the needs of the application.

The SIG additional application layer functions (aalf) is currently evaluating, how the the tasks of CANopen host controllers for network configuration and management, as specified in CiA 302, can be optimized with the tools, provided by CiA 1301. The results are going to be provided in the document series CiA 1302.

B. Device identification by LSS FD

As previously described, CANopen host controllers may have the task to identify unconfigured devices, and to assign them a node number. For this purpose, CANopen host controllers support the LSS manager. In times of CAN FD the CiA SIG LSS FD evaluated, how CANopen host controllers can benefit from CAN FD also with regard to improved LSSs.

The evaluation of the LSSs showed that CAN CC cannot be dispensed with either. The ability to send identical CAN data frames in parallel, at the same time, from different CAN devices, is no longer available in CAN FD. Nevertheless, there was room for improvements.

Improvements by utilizing CAN FD data frames are achieved with the LSS switch state selective service. This service allows a CANopen host controller to selecting exactly one unconfigured CANopen device, by means of its 128 bit “LSS address”. In CAN CC, there was no other possibility than transmitting this address in several segments. Now, this address can be transferred efficiently in one single CAN FD data frame, which is the basis for the updated LSS Switch state selective FD protocol.

In case the LSS address is not known to the CANopen host controller, sometimes the CANopen host controller needs to start a time-consuming scanning process to discover all the LSS addresses of the CANopen devices, attached to the current system. While updating the LSS, the CiA working group was also evaluating, whether there is an option of an improvement of the LSS address scan. As a result of this evaluation, the latest version of CiA 1305 CANopen FD layer setting services and protocols enables unconfigured CANopen device, to inform the CANopen host controller, in segments of four bit (nibble) per CAN CC data frame, about the own LSS address. Having the 128-bit LSS address in mind, a CANopen device can announce its LSS address within 32 consecutive CAN CC data frames, by coding the data in the least significant four bits in the CAN ID, of CBFF-coded data frame. As other unconfigured CANopen devices may do the same task in parallel, the use of the CAN CC data frame is obligatory. Finally, a rapid LSS address scan was introduced as well, based on the CAN CC extended frame format (CEFF). In that case, the least significant 16 bit of the extended CAN CC identifier are used to carry a segment of the unconfigured LSS device's LSS address. This way, just eight CAN CC data frames are sufficient, to announce an entire LSS address. Compared to the early days of LSS, where a CANopen host controller was just enabled to scan Bit-by-Bit for unconfigured devices, this is a significant acceleration for the network start-up time. To offer this non-CAN FD-based improvement, also to CANopen CC host controllers, the SIG LSS FD has already described the option, to map all services, defined by CiA 1305, to CAN CC data frame. The related document CiA 702 is currently in the CiA-internal release process.

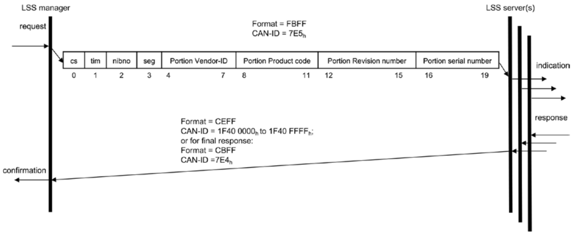

Figure 2: In the improved LSS FD, entire LSS address patterns may be proposed and all unconfigured devices, matching to this pattern, answer by publishing the next 16 bit-part, of their LSS address, coded to the least significant bits of the 29 bit CAN-ID.

C. CANopen bootloader

Prior to the start-up of the system, CANopen host controllers double-check the configuration as well as the software version of the CANopen devices. During that task they may identify that an update of the configuration or the entire CANopen device's firmware is required. With the aim to simplify and generalize the way of running a firmware update via the CAN, the TF Generic bootloader has specified a harmonized procedure. This procedure is provided in CiA 710, which is currently in the CiA-internal release process.

Controllers switch CANopen devices in a finite state automation (FSA) that differentiates between the basic modes of operation, a bootloader mode (BM) and an application mode (AM). When the device starts, the bootloader first switches to the "Bootloader Initializing" status. The existence of a valid application program is checked using checksum verification. If no valid program is found or starting a program is not required, the device switches to CANopen bootloader mode by default, with the settings specified by data object 1F59h. After the initialization operations, the device enters the "Initialization" state, where the setup operations, including CAN controller initialization and bit rate configuration, are performed in accordance with CiA 1301.

In Bootloader (BM) mode, the device verifies the user authentication. Legal CANopen host controllers have therefore the possibility to identify themselves to the CANopen device in bootloader mode. The successful identification initiates the CANopen device being ready to accept new application programs or configuration data. Therefore, the CANopen device enters the BM Allow Application Download state, where it waits to receive a new application program. Prior to the transfer, the CANopen host controller can learn the CANopen device's attributes such as flash status and flash operation times. This enables CANopen host controllers to adapt their behavior and in particular their internal time-outs accordingly. After successful program reception, the CANopen host controller can initiate the CANopen device to leave the bootloader mode, and to start the (typically) the new application program.

When the new application program has been started, the device runs in Application Mode (AM). If a transition back to bootloader mode is required, e.g. to modify the configuration or the entire application program, the CANopen host controller has to pass the security check again, prior to initiate switching back to bootloader mode.
Overall, CiA 710 enables CANopen host controllers to orchestrate the operating states of the device, to ensure secure firmware updates, and to maintain the integrity and reliability of the system operation. To increase the availability of the devices in the systems, a roll-back resp. recovery function has been considered. In case of an error scenario during the firmware update, the device is not lost but will start with a preconfigured default application program.

D. Safety and Security

One of the big advantages of communication systems that are based on broadcast communication is that a central host controller is not necessarily required. Thus, in case a single device detects and communicates a safety-relevant issue, other devices connected to the network can directly react on this information, without the detour via a CANopen host controller. CANopen safety offers exactly such opportunities to accelerate the fault reaction. Nevertheless, CANopen host controllers may monitor the current status of the application, with regard to cyber security and functional safety, and may react on erroneous tendencies on an early stage.

With regard to functional safety, CANopen host controllers can use the CANopen Safety protocol, to communicate safety-relevant data via CAN, according to IEC 61508. The CANopen safety protocol was developed under patronage of CAN in Automation (CiA), and published as a European standard EN 50325-5. Although CANopen safety controllers are not defined in particular, the EN is also feasible for CANopen host controllers supporting CANopen safety. Whether CANopen safety meets still the requirements, derived from the new European machine directive, is currently evaluated by the CiA SIG Functional safety.

CANopen does not offer a standardized solution for cybersecurity. Thus, CANopen host controllers have to choose from various CAN-based security solutions on the market [1], [2]. Those solutions have different attack scenarios in mind, and meet therefore different requirements. The CiA IG “Safety and security” specifies generic security options for CAN CC- and CAN FD-based protocols. It is intended that the CiA 720 document series – currently under development – will specify a cybersecurity higher-layer add-on function. Being of general interest, the approach is pursued by defining generic objects, parameters, and roles required in such a way, that they can be mapped for example to CANopen CC and CANopen FD.

V. Summary

Embedded host controllers are the central element of CANopen-based applications. They are responsible for the proper configuration, commissioning and operation of embedded applications. In addition, they can allow a connection to cloud applications and are responsible for energy-efficient operation of the application. CANopen specifications provide harmonized tools to make CANopen controls of complex tasks, in multi-vendor systems, as simple and efficient as possible. The comprehensive functionalities of the CANopen specifications, in particular CiA 302 for CANopen CC, have made a significant contribution to facilitating network initialization, device configuration and the initiation of counter measures, in case of error scenarios.

With the transition to CANopen FD, new functions and options are available to meet the increasing requirements of embedded network management. CANopen FD brings significant improvements, such as expanded data field sizes and dynamic cross-communication capabilities, enabling host controllers to efficiently handle more complex tasks, in a reduced time.
The functionalities introduced in CiA 309,

which enable host controllers to act as gateways to IoT and web-based services, thereby improving connectivity and accessibility, are also supported by CANopen FD. In particular, the CANopen FD USDO is characterized by its ability to transfer any data size within dynamically changeable communication relationships and thus significantly increase data throughput and system performance.

Looking to the future, ongoing standardization efforts such as CiA 710 for the CANopen bootloader, applicable in CANopen CC and FD applications, promise to further rationalize firmware updates and ensure, for example, that already installed systems can be continuously improved with regard to identified weaknesses in terms of cybersecurity. Given the continuous development of the CANopen specifications with particular attention to interoperability in multi-vendor systems, efficiency, and flexibility, from which host controllers in particular benefit, the future of embedded CAN systems appears promising.

Bibliography

[1] CAN in Automation, Bernd Elend, Tony Adamson, NXP, CAN security enhancing CAN transceivers, Proceedings of the 17th international CAN conference

[2] CAN in Automation, Olaf Pfeiffer, Christian Keydel, Embedded Systems Academy, Proceedings of the 17th international CAN conference

[3] CiA 301, CANopen application layer and communication profile, Version 4.2

[4] CiA 302, CANopen additional application layer functions

[5] CiA 306, CANopen electronic device description

[6] CiA 309, CANopen access from other networks

[7] CiA 320, CANopen services and protocols for sleep and wake-up handling

[8] CiA 458, CANopen device profile for energy measurements

[9] CiA WD 702, Usage of CiA 1305 LSS FD services in CANopen CC environment

[10] CiA WD 710, Generic CANopen bootloader

[11] CiA 1301, CANopen FD application layer and communication profile, Version 1.0

[12] CiA 1305, CANopen FD layer setting services and protocols

[13] EN 50325-5, Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces - Part 5: Functional safety communication based on EN 50325-4

[14] IEC 61508, Functional safety of electrical/electronic/programmable electronic safety-related systems

[15] ISO 11898-1:2015, Road vehicles – Controller area network – Part 1: Data link layer and physical signalling

[16] ISO 11898-2:2016, Road vehicles – Controller are network – Part 2: High-speed medium access unit

[17] network

Reiner Zitzmann
CAN in Automation
Kontumazgarten 3
DE-90429 Nuremberg
headquarters@can-cia.org
www.can-cia.org