December 2016

CAN Newsletter

Hardware + Software + Tools + Engineering

Sensors

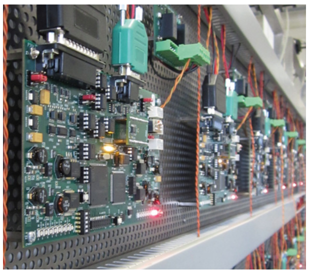
3D Lidar for mobile machinery

Autonomous mobile work machines need the capability of sensing and mapping the surrounding area. Finnish researchers developed 3D Lidar, based on a 2D laser scanner and electric motor drive that rotates the scanner.

Table of contents

Sensors

CANopen



CAN FD

Imprint

Publisher
CAN in Automation GmbH
Kontumazgarten 3
DE-90429 Nuremberg

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

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

CEO Holger Zeltwanger
AG Nürnberg 24338

Downloads September issue:
(retrieved November 8, 2016)
4000 full magazine

Editors
pr@can-cia.org

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

Layout
Nickel Plankermann

Media consultants
Gisela Scheib
(responsible according
to the press law)
Meng Xie-Buchert

Distribution manager
Meng Xie-Buchert

25th anniversary

With the 1/17 issue, CAN in Automation is going to celebrate its 25th anniversary, which means that we would like to create a special magazine. For this special purpose, we are planning to print the magazine again. It would be great if you could support us with application reports, press releases, articles etc. about CAN/CANopen/J1939/Devicenet/Isobus – anything that provides an outlook to the future of CAN. Maybe you have something to say about CAN regarding service robots, security, car hacking, autonomous driving, e-bikes and so on?

Robotic cars for the future

It is not science fiction anymore: a car with an adaptive and modular design that fulfills the personal requirements of a driver better than a usual car. The EO smart connecting cars aim to solve urban traffic problems.

Figure 1 & 2: From concept to reality: EO smart connecting car 1 (left) and 2 (right) developed by DFKI-RIC (Photo: DFKI)

The Robotics Innovation Center (RIC) at DFKI (Deutsches Forschungszentrum fuer Kuenstliche Intelligenz/German Research Center for Artificial Intelligence) in Bremen has shown its experience with a range of projects involving self-developed, upgraded, modular, and reconfigurable vehicle platforms in the electric mobility field that aim to meet the needs of urban life. The EO smart connecting car (EOscc) concept vehicle family is known for its morphological adaptation and its modular construction concept for a spectrum of specific applications, which makes it a pioneer in its field. “EO” means “I go” in Latin. EOscc1 and EOscc2 are designed as micro-car sized electric robotic vehicles, where complex robot design requirements have to be met, among them extended mechanical functionality demands, as well as vehicle requirements, for example high power capability and robustness. With the help of rapid development methods, optimization, and modular configurable approaches, EOscc2 and EOscc1 are vehicles that can meet individual mobility requirements for everyday life. Both cars are four wheel-driving electric vehicles with x-by wire (steer-brake-throttle) control, which reaches extended maneuverability through its suspension/axle design and decentralized power train (wheel hub motors and brushless DC motor controllers). With these features, the vehicles eliminate the problems of urban traffic, like shortage of parking spaces, maneuverability in extreme traffic situations, and more. With the coupling mechanism for Car-2-Car, Car-2-Extender, or Car-2-Infrastructure (charging station, rental station), the cars reach high modularity. Modules like the range extender and the pick-up module allow vehicle extension up to the road train mode and are based on hardware and software configurability. They enable a higher efficiency and individuality of applications.

Because of these features, the electric robotic systems are quite complex. In order to cope with this complexity, these issues are addressed through the design and the development. With this perspective, EOscc1 was constructed with a total of 24 independently-controllable off-the-shelf linear electric actuators, which adjust the mechanical body part position for steering (from double Ackermann up to sideways driving), lifting (from adjusting the height of wheels up to changing the curve tilting of the vehicle), and folding of the car individually. Each linear electric actuator, for example the brushless DC motor controller of the wheel hub motors, is controlled separately by an industrial central PC (PC/104) via four different CAN networks (front axle, rear axle, morphology, control and power electronics). This achieves a higher modularity and flexibility in a basic physical bus topology, as well as easier system development, and less installation and testing effort.

EOscc2 is the second generation of the DFKI modular concept electric robot vehicles with optimized and modified ▷

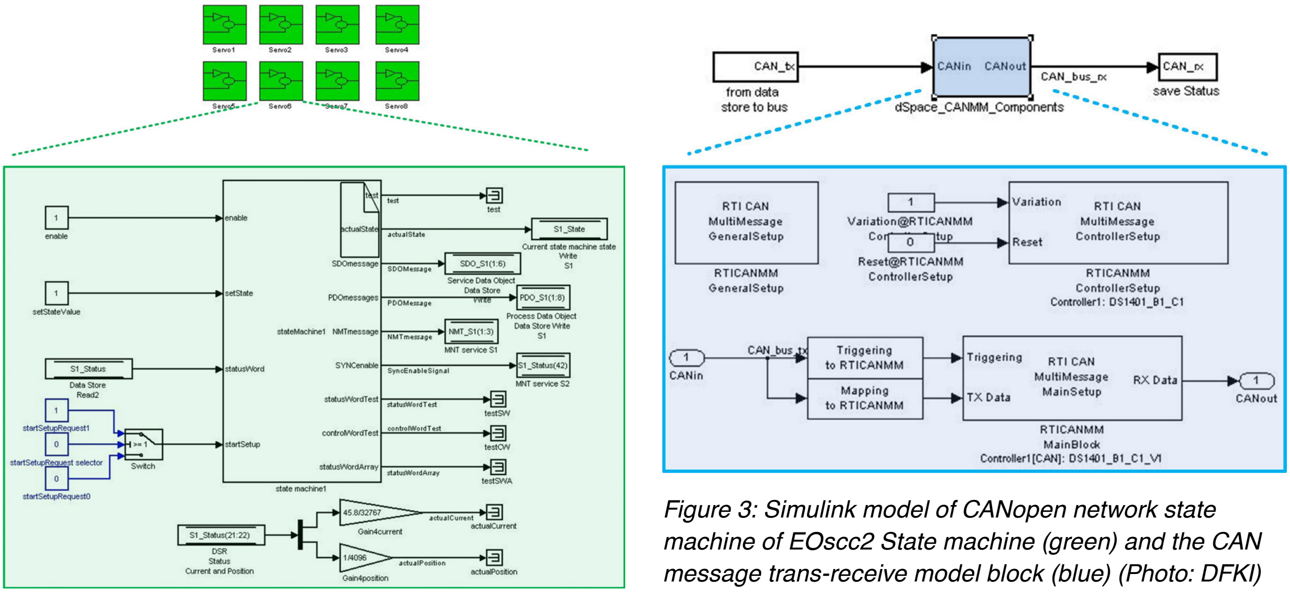


Figure 3: Simulink model of CANopen network state machine of EOscc2 State machine (green) and the CAN message trans-receive model block (blue) (Photo: DFKI)

suspension and body design for robustness and extended functionality, which aim to solve urban traffic problems and achieve autonomous driving. During the development of the control layer of the EOscc2, we kept the whole system as clearly arranged as possible, aiming at a feasible system implementation with our experience from EOscc1. Therefore, the control layers “high-level” (perception and planning for robotic behavior on Robot Construction Kit framework) and “mid/low-level” (actuation of whole hardware components) of the EOscc2 are separated, unlike the control layers of the EOssc1.

During the development of the EOscc2, we decided to use a linear actuator designed and constructed in-house, as we found no off-the-shelf linear actuator solution that fulfilled our requirements. To build such a linear actuator, an industrial synchronous servomotor with integrated powerful driver electronics and ball screw or acme thread type spindle gear were combined. The actuators are built with servo control electronics and communicate via the CANopen network, which is a sophisticated and secure communication standard for distributed industrial automation systems. The selected servomotors of the EOscc2

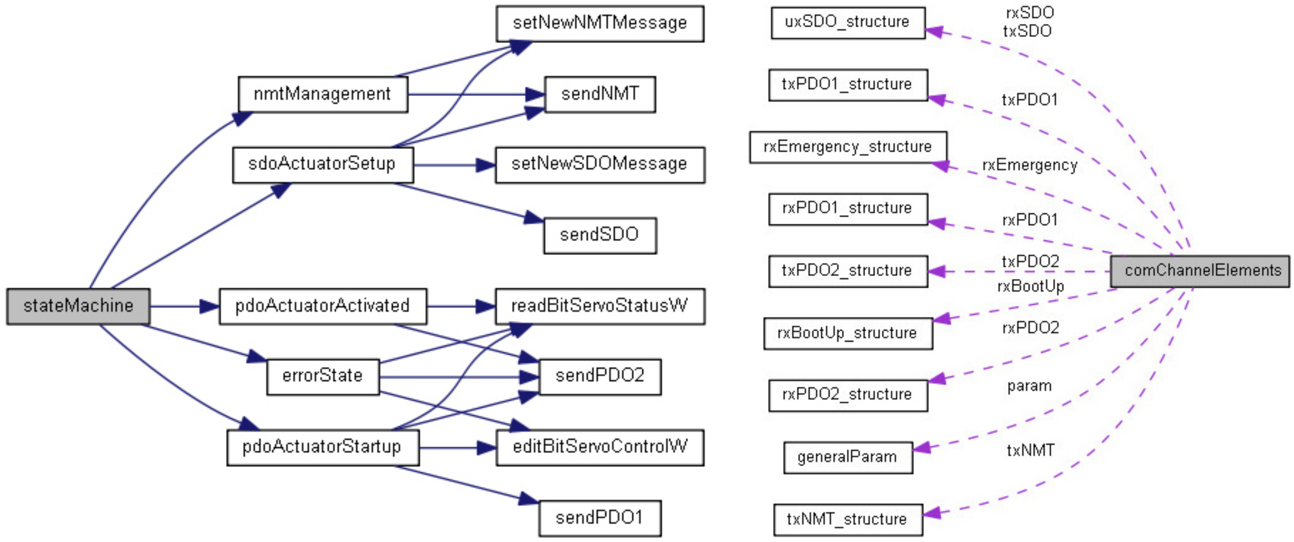


Figure 4: The components of the CANopen state machine routine of EOscc2 (Photo: DFKI)

actuators gave the system development the necessary flexibility for different tasks, for example steering (between +32° to -92°), lifting (in total 16 cm), folding (to decrease the parking space from 4,32 m² to 2,84 m²) or the manipulation of other body parts. They are used as servomotors with matching physical characteristic and CANopen network, which enables the addition of servos up to 127 network nodes without a huge programming effort on the control software. The robust CANopen communication, which is based on the CiA 301 standard, as well as the adjustable device profiles and different motor control functions increases the dynamics of the robot system. Especially some features of CANopen, such as standardized device function and parameter description (Object Dictionary) keep the system straightforward and at the same time modular and flexible. The “producer-consumer model” without any additional protocol overhead enables efficient transmission, so that the message of a node is transmitted to other CANopen nodes like a broadcast message. This solves the complex synchronized critical control tasks of the robotic vehicle.

To realize the control of the whole CANopen network, we started to develop rapid control prototyping (RCP) methods and created a model-based experimental control logic of CANopen network according to CiA 402 in a Matlab/Simulink environment. Later, this network management and state-machine control logic subsystem (see the green Simulink model in Figure 3) was connected to the mid- and low-level control layer (steering and lifting actuator control, steer/drive/brake-by-wire, communication, and user interfaces) and the kinematic models of the robotic system to control all network nodes as well as other vehicle control objects on different CAN networks. Because of its modularity and scalability, the same software module could be re-used within the main model controlling the whole car. Afterwards, the whole system was tested in a Hardware-in-the-loop (HIL) platform. Thus, the RCP unit could iteratively be used for device control and parameter tuning starting from the early stages of the development.

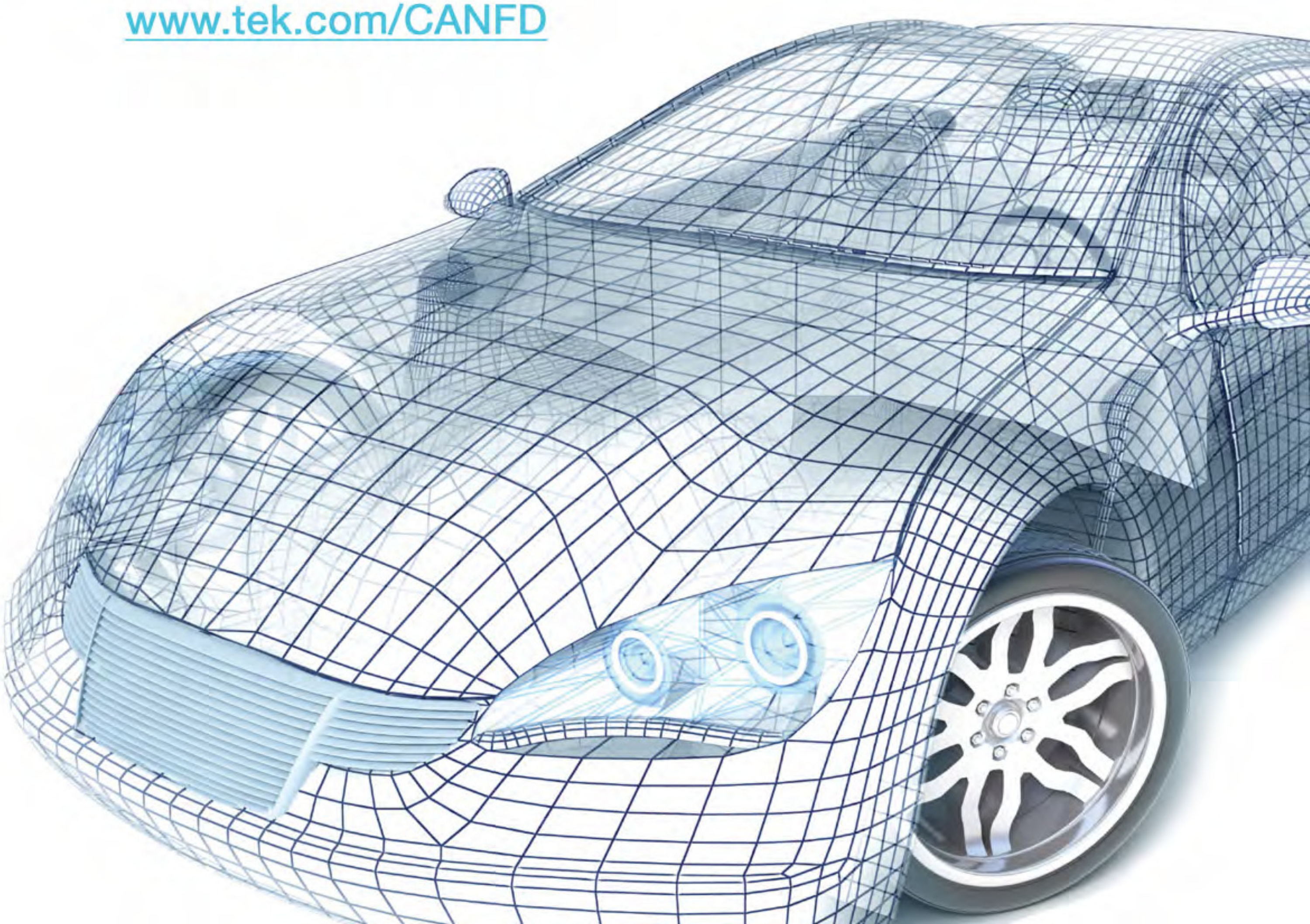
The last step was adapting the protocols for the microcontroller (32-bit ARM Cortex-M) of the vehicle control unit (VCU), reusing some parts of the RCP model and code, which were already written in C (see Figure 4). Hence, the VCU micro-controller took over the CANopen network management tasks (NMT) within different subroutines, for example the setup of servo motors (via Service Data Objects (SDO) direct access to Object Dictionary (OBJ) and re-mapping of Process Data Objects (PDO) parameter), as well as observing and controlling device states (status and control word management) with 500 kbit/s CAN synchronized broadcast communication.

The whole development of the CANopen network control and device management was successfully transferred to another platform with the aim to upgrade four on-the-shelf MIA electric vehicles in order to manipulate the steering and the braking systems of the vehicle for autonomous driving in the Dabrem project (Dalian Bremen Electric Mobility).

This work was performed in the subproject ”Innovative Technologies Electromobility” of the main project ”Model Region Electric Mobility” – Module 2 ”Intelligent Integration of Electric Mobility” (grant number 03ME0400G). It was also part of the project “Dalian-Bremen Electric Mobility”. The work was funded by the German Federal Ministry of Transport and Digital Infrastructure (grant number 03EM0404A). The program coordination was carried out by the National Organization Hydrogen and Fuel Cell Technology.

Author

Mehmed Yüksel
Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
German Research Center for Artificial Intelligence - Robotics Innovation Center
mehmed.yueksel@dfki.de
www.dfki.de/robotik



Managing CANopen devices

FDT Group standardizes device management for fieldbuses regardless of manufacturer. Which components and mechanisms are required to implement the FDT interface in a PLC programming environment such as Codesys?

(Photo: Schneider Electric)

The FDT technology has its roots in process automation. Due to the variety of fieldbuses found in factory automation, this technology is also useful in this exact environment for standardized device management. This is because the PLC programming systems which are common in this environment – such as Codesys – can ensure significantly better device integration if FDT is used. A software component, referred to as the Frame Common Component, is available for implementing the FDT interfaces in an FDT application – including an FDT frame application. This component, which is available from M&M Software and is certified by the FDT Group, is used as the base component. Using this component ensures that implementation conforms to the specification and improves interoperability with the various DTMs (Device Type Manager).

The Frame Common Component contains all of the interfaces which are required in accordance with the FDT specification to develop an FDT application. This kind of application can perform a wide range of functions, for example it can be used as a stand-alone tool for device configuration, a diagnostics tool, or an asset management tool. In Figure 2, the FDT application is represented by the outer frame. This is where all the functions of each application are implemented. The inner part shows the base component (fdt Container component). Interfaces connecting the two elements are located between the application and the base component. The direction of the arrows shows whether the interface calls up the base component or the application. The database adapter (DB Adapter) can be seen in the lower part of the application. It forms the adaptation layer for the application-specific database. The functions shown in the database adapter (for example Project Record or DTM Instance Data) are placeholders for programming database access.

The base component itself contains functions for managing the DTMs (DTM Catalog) and the application ▷

project (Frame Project). The system topology, the associated DTMs (DTM List) and a proxy form part of the project. This proxy separates abstract online call-ups into individual FDT function call-ups – for example "go offline". The instantiated DTM itself runs in the DTM Container component.

Codesys plugin concept

Codesys is a software suite for automation technology. The basis for this is the IEC-61131-3 programming tool. This makes it possible to integrate additional functions using custom-designed plugins such as new menus, editors, etc. The FDT base component is also embedded into the programming tool with this mechanism. Figure 2 shows a detail of the Codesys architecture, including the FDT components. The project structure, as shown in Codesys, should be considered in parallel to this (Codesys device tree). The FDT Integration Plugin contains the FDT base component for managing an FDT project with the associated DTMs.

The architecture illustrated in Figure 3 shows the integration of CANopen devices. It is possible to use devices both with and without DTM. The CANopen Master is represented by a DTM which is used to configure CANopen. It is also the communication interface with the CANopen devices. The IFdt Communication interface, which is used by a device DTM to exchange data with its slave (DTM for Slave 2), is used for this purpose. The CANopen Master DTM converts the data from the device DTM into CANopen-specific messages and sends these to the device. The response from the device is processed accordingly in reverse order.

Device management makes it possible to mix components which do not require an explicit DTM due to their simplicity with devices with increased functionality and whose full scope of functions can be used thanks to the DTM (see Figure 4 with a DTM for a servo drive providing an oscilloscope to observe different parameters). Codesys contains a general XML device description for all devices as well as the associated fieldbus device file which is EDS for CANopen in this case. The device description contains all the information relating to the device, its parameters, and communication options. If a device with DTM is used, the DTM is transferred from the DTM catalog to the project, and the associated XML description is generated automatically. The system receives the required data via the IDtm ▷

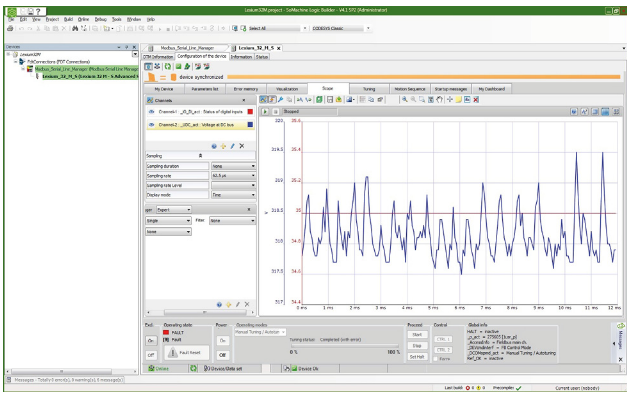


Figure 1: DTM for a servo drive providing an oscilloscope to observe different parameters (Photo: Schneider Electric)

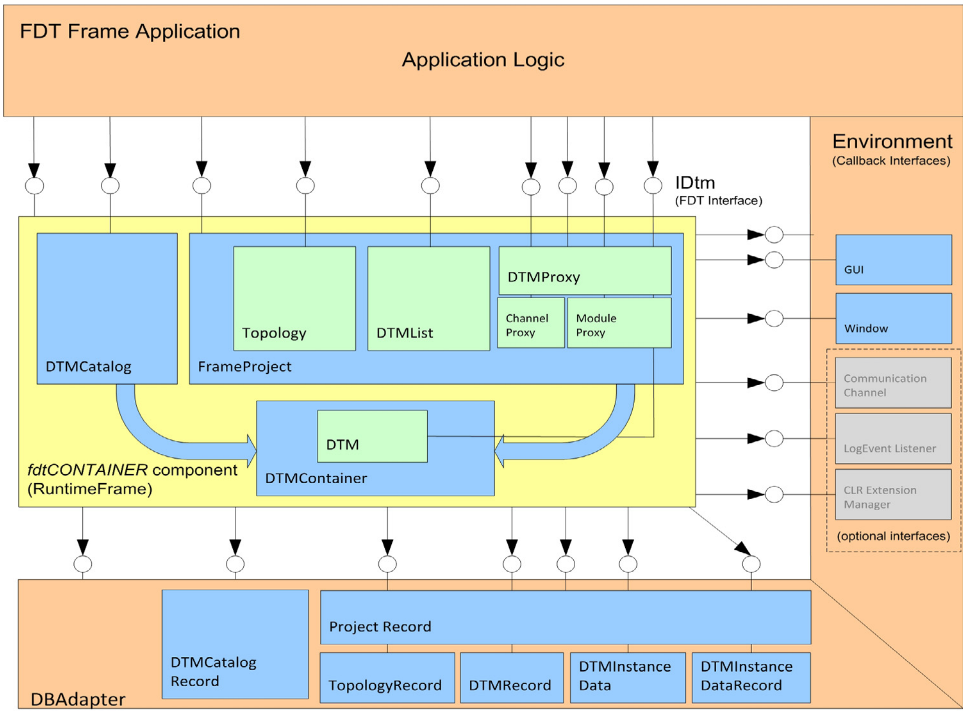


Figure 2: The FDT application is represented by the outer frame (Photo: Schneider Electric)

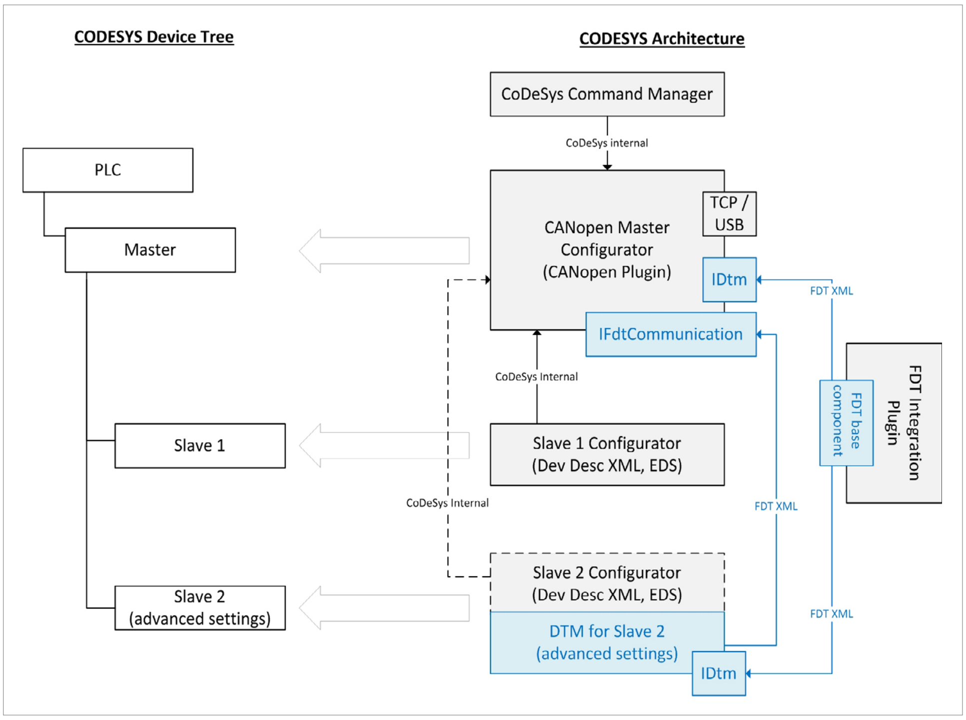


Figure 3: Integration of CANopen devices (Photo: Schneider Electric)

interface. Information relating to the manufacturer, device type, and version data is made available in this way, as well as other information. Frequency converters are an example of a device with a wide range of functions. In this case, it is advantageous to provide each device range with a DTM that covers the entire scope of this range.

The CANopen Master Configurator enables convenient configuration of the fieldbus. This includes parameters for the fieldbus itself such as bit-rate or heartbeat (monitoring time). This DTM also assigns the addresses for the devices (Node ID) and maps PDOs (Process Data Objects). The latter establishes the connection between the PDOs and the application objects in the object dictionary. The Master Configurator exchanges this data with the device DTMs via the IDtm interface (Set Parameters). The Master Configurator is also notified when data is changed by the device DTM. They can then request these changes via the interface (Get Parameters). After configuration is complete, the Master Configurator checks the consistency of the data in the CANopen system. The CANopen data is downloaded to the Master and the devices and used during runtime for starting up the system, setting the fieldbus parameters and exchanging data.

Generating process data

Process data is the data transferred between the PLC and the slaves during the fieldbus runtime. The description of the process data, which is defined by means of the device DTMs, is made available at a corresponding interface (process channels). In this case, the information from the EDS (Electronic Device Description) is not used as it is insufficient for this purpose.

This procedure is helpful particularly for modular devices with a variable configuration. The PDOs (Process Data Objects) are automatically generated for this process data. Existing variables can be assigned to the process data or redefined in the device DTM. The PLC tool receives the variables from the device DTM and makes them available to the PLC program in a final step, which then allows them to be used directly by the PLC programmer for the purpose of further processing.

To sum up, using FDT technology in Codesys provides programmers with a wide range of options for device management. The functions and flexibility by far exceed what is possible as a result of using the usual description files (EDS, GSD). For instance, ▷

What is FDT technology?

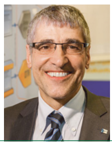
FDT standardizes device management for field devices in one tool. Device management is standardized regardless of the device manufacturer and the fieldbus/network used. This means that every device in an FDT tool can be configured, operated, and maintained by standardized interfaces – regardless of manufacturer, type or communication protocol. The three key elements of FDT are:

The FDT interface – the standard for device management: The FDT interface is the specification which describes standardized data exchange between devices and the control system, or the engineering and asset management tools.

The DTM (device driver): The DTM (Device Type Manager) provides a standardized structure for accessing the device parameters as well as configuring and operating devices and performing fault diagnostics. DTMs range from a simple graphical user interface for parameterization to a sophisticated application which can handle complex real-time calculations for diagnostics and maintenance. DTMs are divided into three categories: device DTM (is supplied by the device manufacturer, represents the entire logic and parameterization of a device, creates a standard interface with the FDT frame application, can be used in any FDT frame application, is based on the DTM style guide), communication DTM (represents communication components such as PC network cards and couplers), and gateway DTM (represents components that connect two different networks/fieldbuses).

The FDT frame application (host system): The frame application is a software program which integrates device, communication, and gateway DTMs from various manufacturers and for different networks/fieldbuses. The frame application offers: a shared, standardized environment, user management, DTM management, data management, network configuration, and navigation.

devices from different manufacturers can be integrated into one tool in the same way and access to the devices is ensured beyond the scope of fieldbus hierarchies. With their graphical user interface, DTMs make accessing the relevant device simple and also offer device-specific diagnostics functions which enable rapid fault detection in the event of maintenance.



Authors

Manfred Brill
Nuno Barral
Schneider Electric Automation
manfred.brill@schneider-electric.de
nuno.barral@schneider-electric.de
www.schneider-electric.de

Reliable communication for mobile machines

Communication between the automation components within mobile machines is becoming more important – and so is the communication with the machine from the outside.

(Photos: ifm)

Mobile machines, for example in the construction sector, present particular challenges when it comes to applied automation technology. This includes difficulties related to dust, water, and vibrations as well as particularly high or low temperatures. Apart from these problems that we all know all too well, another factor is becoming more and more important: The communication between the automation components within the machine and the communication with the machine from the outside. Machine automation requires perfect interaction of controllers, sensors, and actuators as well as devices for user interaction. The communication between all components must be absolutely reliable. For mobile machines, such as construction, agricultural, and municipal vehicles, CAN, which was initially developed by the automotive industry, has become the predominant communication protocol.

In order to enable analog sensor signal transmission via the CAN network, usually I/O modules are used. For the mobile machine segment, the sensor manufacturer ifm electronic offers Compact Modules and Smart Modules that can be connected to various sensors. Apart from inputs for analog and digital signals, the I/O modules have outputs that can be used to control actuators. Data transmission to the controller takes places via the CANopen protocol. ▷

Figure 1: The sensor signals are routed to the controller via CAN using I/O modules (Photo: ifm)

CAN is ideal for sensor data transmission within the machine. The transmitted data volume is relatively small, and the distances between the individual components are short, so that the performance of CAN is sufficient. Today, however, it is more and more often required to also integrate the machine into a more wide-ranging network and to be able to access the machine via the Internet. For mobile machines this is significantly more complex than for stationary machines, for example, in production processes. With the CAN Remote radio gateway, ifm electronic offers a possibility to access the CAN network of a mobile machine via a mobile network connection. The solution includes a complete hardware and software package that makes it possible to implement remote maintenance and other functions. Since a GPS receiver is also integrated in the units, it is also possible to determine the location of the machine via the CAN Remote system. Especially for construction machine manufacturers the possibility of carrying out remote maintenance or other services over long distances is very important. Often those machines are operated all over the world and in remote places. Therefore, CAN Remote can reduce travel costs for servicing.

Figure 3: CAN Remote makes it possible to access CAN in a mobile machine via mobile network connection (Photo: ifm)

For mobile machines, not only communication with the outside world is important. Also for communication within the machine there are technical innovations offering the user various advantages.

Figure 2: ifm electronic offers a complete product portfolio for mobile machine automation including sensors, controllers and dialog units for visualization and operation (Photo: ifm)

Figure 4: ifm electronic offers a complete product portfolio for mobile machine automation including sensors, controllers and dialog units for visualization and operation (Photo: ifm)

This includes, for example, IO-Link integration of sensors. This standardized digital communication protocol offers many functions that would be impossible with conventional analog transmission. One of them is, for example, simple sensor diagnostics. Via IO-Link, the higher-level controller can be directly informed about any errors. This makes trouble-shooting easier. Also when replacing a sensor, IO-Link shows its advantages. No complicated configuration and readjustment of the sensor is required since the IO-Link master can transmit the configuration data directly to the sensor. In addition, this type of sensor data transmission is much more immune to interference than analog transmission. IO-Link has already become an industry standard. The interest that construction machine manufacturers take in the system is currently also increasing – a development that will surely intensify in the years to come. This is why ifm has provided all new sensors with an IO-Link interface. Additionally Ethernet-based systems are becoming more and more popular for mobile machines. But it is still undecided which protocol will become the absolute standard. To guarantee their customers sustainable options for the future, ifm electronic integrates Ethernet TCP/IP in the units and is thereby prepared for all developments.

The demands on mobile machine automation are becoming more and more complex. Many sensors are required that usually communicate with the controller via CAN. Apart from controllers and dialog units for visualization and operation, today often connections to a remote maintenance solution are needed. Mobile machine builders benefit from the fact that ifm offers them all components from a single source.

Author

Dietmar Brüss
ifm ecomatic
dietmar.bruess@ifm.com
www.ifm.com

3D Lidar for mobile machinery

Autonomous mobile work machines need the capability of sensing and mapping the surrounding area. Finnish researchers have developed 3D Lidar based on a 2D laser scanner and electric motor drive that rotates the scanner.

Autonomous mobile work machines require accurate maps of their surroundings in order to be able to perform the required tasks efficiently and especially safely. The machines need first of all rough information of their position at the work site, where an accuracy of a few meters is often acceptable. This can be provided to the machine with GPS-based navigation. However, in addition they need high-resolution information – accurate to a few centimeters – of their position in relation to the nearby objects and environment. As the machines are constantly moving, this information needs to be constantly updated.

Some commercial solutions of 3D Lidars can be found on the market, for example from manufacturers such as Sick [6] and Velodyne [10], but in many cases these are quite expensive, especially for outdoor applications. Therefore, there is a need for a robust low cost 3D Lidar, especially for use in research. We present a solution for a low cost 3D Lidar based on a 2D laser scanner, an electric motor drive rotating the 2D scanner, and data fusion with the navigation system of the mobile machine. The 2D laser scanner provides range and intensity data from the measured plane. The controller of the electric motor drive provides the rotation angle and rotation speed of the laser scanner. By combining these measurement data with the navigation data one can create point clouds that can be used for sensing and mapping the environment of an autonomous mobile machine. In this article, the mechanics and control system structure of the servo are presented in detail to make building similar types of systems easier for the interested reader. The developed hardware described in this article was made for the laser mapping of the autonomous mobile machine used in the GIM project [2]. The rotating laser scanner scans the fore field of the machine. The laser map is used among other types of maps for path planning and obstacle avoidance purposes.

Control system of an autonomous mobile machine

The work machine, for which the mapping system was developed, is a modified version of a multi-purpose wheel loader. The frame of the machine is original, but the control system, electronics, and hydraulics have been changed to enable researching autonomous operations. The control system architecture of this autonomous machine is illustrated in Figure 1. The hardware devices are located physically in two different locations. Visualization and operator computers are off-board computers and those are connected through a network switch and over WLAN to an on-board network switch. On-board computers and peripheral devices are also connected to the on-board switch. The on-board computers and peripheral devices follow the device architecture illustrated in Figure 1.

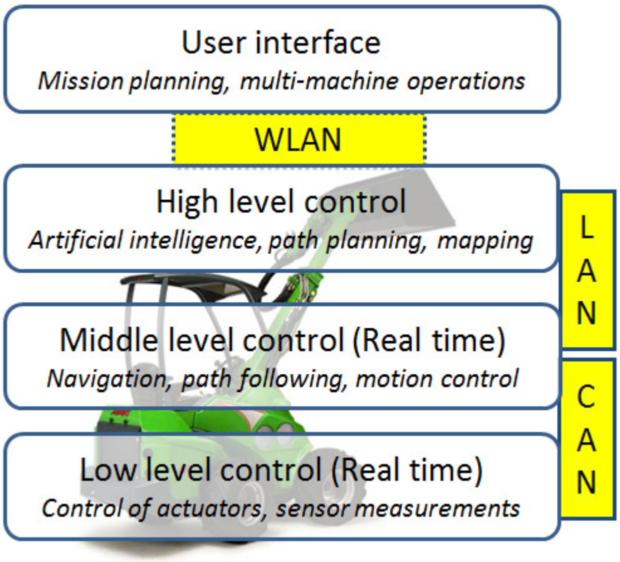


Figure 1: Control system architecture of autonomous mobile machine [3] (Photo: TUT)

The low-level control device architecture consists of a control of the actuators, such as hydrostatic drive pump, diesel engine, and hydraulic valves of the machine. It also takes care of the data logging of the inertial measurement unit (IMU), central joint resolver, pressure, and some other sensors. The low-level control is based on six Epec embedded vehicle computers, which communicate with the middle level control through four CAN networks. The implementation of CAN followed the CANopen standardization, which enables fluent CAN network management and consistency handling.

The middle level control, including navigation, path planning, and network interfaces with real-time kinematic global navigation satellite system (RTK-GNSS) was implemented on an industrial embedded PC using Matlab/xPC Target as an operating system. The control of the servo – including the electric motor drive and the encoder measurement – was also implemented in this level via CAN. The high-level control of device architecture is also implemented on an industrial embedded PC. This PC communicates with the middle level using the UDP protocol. This protocol is also used for the data transfer from the 2D laser scanner to the middle level ▷

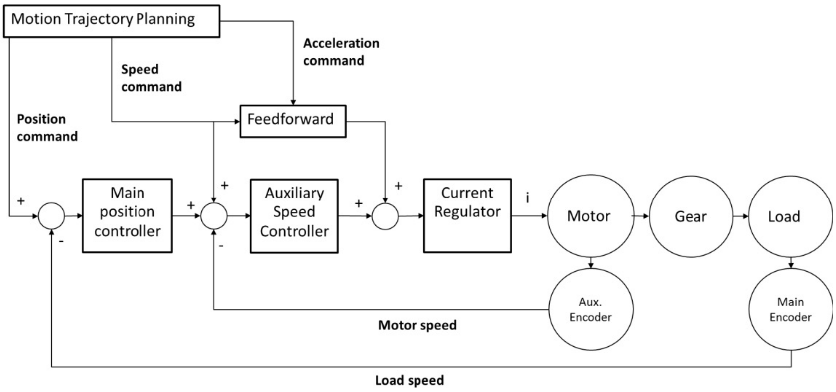


Figure 2: Control system with two control loops (Photo: TUT)

control to ensure a real-time data transfer of the laser scanner data.

Electric motor drives and 2D laser scanners

In general, two different types of DC motor exist: Brushed direct current (DC) motors and brushless direct current (BLDC) motors are the most typical motors. BLDC motors are typically driven by three phase conductors and phase voltages are generated by the braking voltage of the intermediate circuit. In accurate position and speed electric motor drives, BLDC motors are more common. [4] Thus, a BLDC motor was also used in this case. A ceramic planetary gear was used for the reduction of the output rotation speed and to increase the output torque.

Feedback sensors

The accurate and not (or in practice minimum and constantly) delayed measurement of the rotation angle is essential to later enable data fusion with laser scanner range data for creating a 3D point cloud data. This is more important than for example an accuracy of the position or speed control of the servo because in these cases a small error or delay is not reflected to in the 3D point cloud calculation.

DC motors typically use a built-in brushed commutator for commutation but the selected BLDC motor requires an external sensor to sense the angle and speed of the rotor for commutation. The commutation type depends on the sensor type and it can be sensor-less commutation, six-step commutation or sinusoidal commutation. Sensor-less commutation doesn't use sensors at all, six-step commutation uses Hall sensors, and sine commutation uses for example incremental encoders for commutation. Sine commutation is the most recommended commutation type if the application requires a constant torque generation over the whole rotation speed range [7]. Thus, an external incremental encoder was selected for measuring the speed of the electric motor for speed control purposes. The measurement of the incremental encoder is a relative measurement, referenced to a certain reference rotation angle of

Table 1. Properties of a Sick LMS111-10100 laser scanner

Property	Value
Angle resolution	0.25° / 0.5°
Output frequency	25 Hz / 50 Hz
Operating range	max 20 m
Field of view (FOV)	max 270°
Systematic error	± 30 mm
Statistical error	Typical 12 mm

Table 2: Parameters of the electric motor [7]

Property	Value
Nominal voltage	24 V
No load speed	9250 rpm
No load current	123 mA
Nominal speed	7230 rpm
Nominal torque	33,6 mNm
Nominal current	1,48 A
Speed / torque gradient	59,1 rpm / mNm
Speed constant	393 rpm / V
Torque constant	24,3 mNm / A

rotor. It can't be directly used as a rotation angle measurement because every time the sensor is started it has to be calibrated, i.e. by finding the zero position. Therefore, an external absolute encoder was selected for measuring the rotation angle despite the fact that absolute encoders are typically more expensive than incremental encoders because the internal structure of absolute encoders is more complex. Because data fusion with laser scanner range data requires the accurate measurement of the laser scanner rotation angle, the rotation angle measurement is made directly from the load side. Thus, inaccuracies resulting from the backlash of the planetary gear can be avoided.

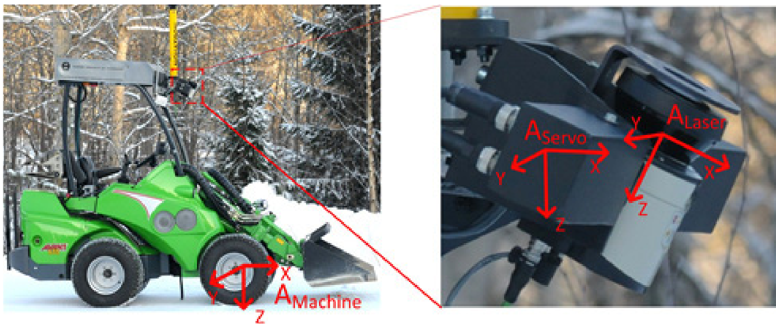


Figure 3: 2D laser scanner (Sick LMS111-10100) installed on an autonomous mobile machine (Photo: TUT)

Control system of the electric motor

In this paper, the control system of the electric motor controller is based on two control loops as illustrated in Figure 2. Two control loops were used because the system includes a planetary gear that brings backlashes to the system under control. Furthermore, backlashes cause delays in control systems and might have an effect on the system stability. The planetary gear is needed because it is hard to precisely estimate the torque that the load requires. The purpose of an auxiliary control loop is to stabilize, define damping and the dynamic behavior of the system. The sensor in the auxiliary control loop is an incremental encoder that is located at the backend of the electric motor. The primary control loop uses an absolute encoder as a feedback sensor and controls the load angle of the rotation.

All controllers in the control system are implemented to the electric motor controller as discrete time controllers. The current regulator is a PI controller with 10-kHz sampling time; the speed controller is implemented as a PI controller with speed and acceleration feed forward. A position controller is implemented as a PID controller with speed and acceleration feed forward. The sampling times for both controllers are 1 kHz. In the control system, the speed feed forward can compensate speed-dependent friction that is caused by bearing among other things. The acceleration feed forward provides more current in cases when one needs a high acceleration or the load inertia is high. The electric motor controller is delivered with ready-made software that can be used for controller tuning. ▷

Table 3: Components of the electric motor drive

Component	Manufacturer	Component type	Details
Electric motor controller	Maxon Motor Ag	EPOS2 50/5	Support for dual loop control
Incremental encoder	Maxon Motor Ag	Encoder MR Type ML	1000 CPT with index channel
Absolute encoder	Scancon Industrial Encoders	SAG-S101G-0016-C06S-PAL	Resolution 16 bit, SSI interface to motor controller
Electric motor	Maxon Motor Ag	EC-max 30	Build-in Hall sensors
Planetary gear	Maxon Motor Ag	Ceramic Planetary Gearhead GP 32 C	Gear ratio 1181:1
Shunt regulator	Maxon Motor Ag	DSR 50/5	-
Choke Module	Maxon Motor Ag	Choke module	-

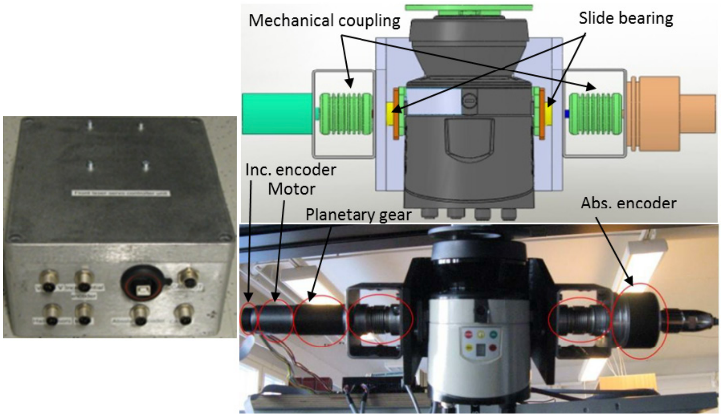


Figure 4: Control unit of electric motor (left) and the schematics and a photo of the 2D laser scanner rotation unit without casing (right) (Photo: TUT)

2D laser scanners

The purpose of the electric motor drive is to continuously rotate a 2D laser scanner. The scanner type – a Sick LMS111 – was predefined for this hardware by the end user. Table 1 presents the most relevant properties of the scanner. The laser scanner and its final installation position on the roof of the autonomous machine are presented in Figure 3. The measurements in Figure 6 illustrate how the position (left) and angular speed (right) behave over one movement cycle. One can clearly see that profiles are quite smooth over time and only a small angular speed ripple exists. The operating principle of the 2D laser scanner is based on the time-of-flight (TOF). The 2D laser scanner gives range and intensity values as measurement values from the measured plane. Measured range values are relative to the rotating mirror of the 2D laser scanner and for every measured point one can also get the intensity value [6].

Hardware and software of the 3D laser scanner

The system level requirements based on simulations were defined as follows: movement range 100°, field of view 140° at the whole movement range, maximum rotation speed 40°/s. The total mass of the load of the electric motor was about 2,3 kg and the calculation of the inertia of the load was based on this mass, the dimensions of the laser scanner, and assuming a homogeneous mass distribution. Based on this data, a Maxon EC-max 30 was chosen as an electric motor ▷

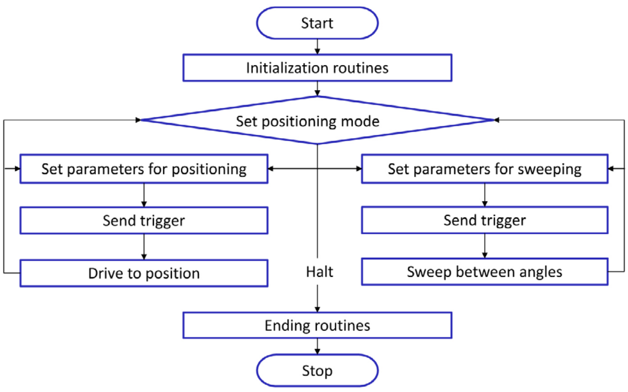


Figure 5: Flow chart of developed software (Photo: TUT)

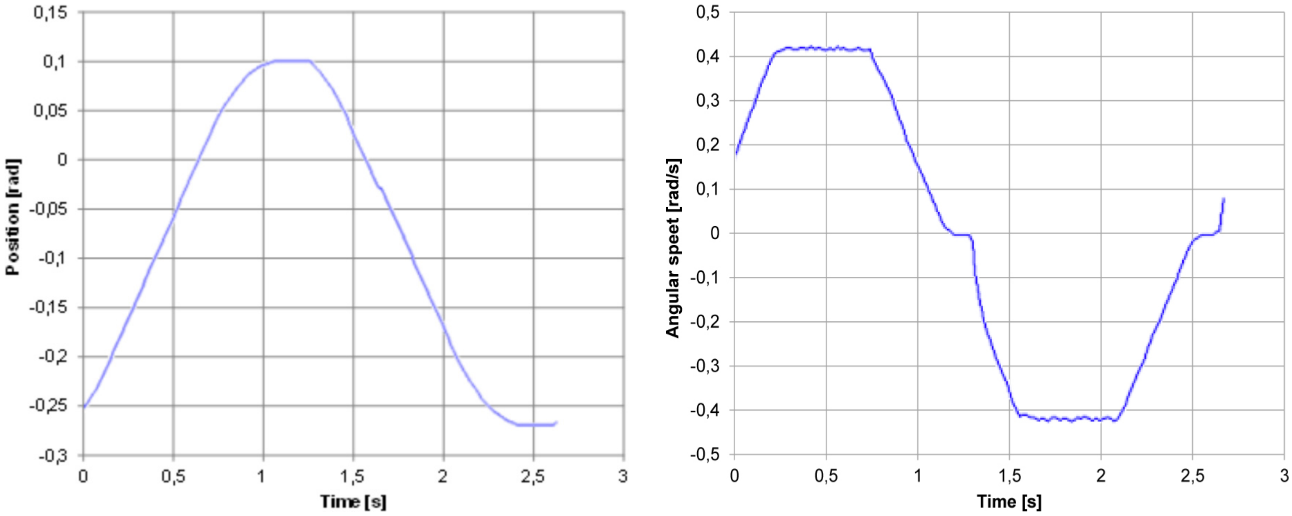


Figure 6: Measured position (left) and angular speed (right) of the electric motor (Photo: TUT)

drive and planetary gear. The electric motor has the parameters shown in Table 2.

The electric motor drive contains a planetary gear, so the electric motor controller must support two control loops. In the rotation speed loop, an incremental encoder with 1000 counts per turn (CPT) and an index channel was used. In the position control loop, a single-turn absolute encoder with 16-bit resolution was selected. The purpose of shunt regulators is to prevent current flowing back to power source. Returning current can cause the power source output voltage to rise. Table 3 presents all components that were needed for the implementation of the electric motor drive.

The mechanics of the hardware consist of two different units. The control unit contains all components that are closely related to the electric motor controller and the rotation unit contains all components that are closely related to the electric motor and sensors. Between these two units exists only wiring for the electric motor and sensors. The case of the electric motor controller did not offer enough protection for a demanding environment; accordingly the case was improved

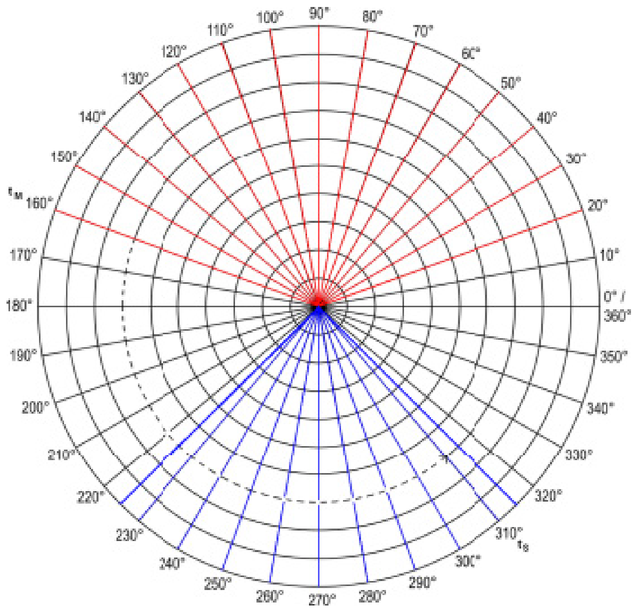


Figure 7: 2D FOV of the laser scanner used in the mapping (red) and the region when laser scanner does the UDP packet sending (blue) (Photo: TUT)

with an additional aluminum case. This aluminum case is easy to move and can be installed for any mobile machine. This control unit with panel connectors is illustrated in Figure 4.

The 2D laser scanner is supported from both sides with slide bearings. These slide bearings are marked yellow in Figure 4. Slide bearings support the mechanical structure so that an axial movement of the 2D laser scanner is prohibited. Another purpose of the slide bearings is to tolerate a radial load. The slide bearings' material is a brass alloy and the axle material against this brass alloy is Hydax 15 steel.

The mechanical structure and manufacturing technique might cause axial misalignment, so components that fit the axial misalignment are needed. Mechanical coupling on both side of the 2D laser scanner makes this axial misalignment fitting, and mechanical coupling forward rotational motion and torque generated by the electric motor and planetary gear. The mechanical coupling fits the axis of the planetary gear and the axis of the mounting patch, seen on the left side of Figure 4. Furthermore, the other mechanical coupling fits the axis of the mounting patch and the axis of the absolute encoder on the right side.

The control system of the mobile machine runs on middle level control. The electric motor drive and 2D laser scanner are controlled by a smaller control unit seen in Figure 4, which was integrated in an existing control system. The execution of the developed software follows the flow chart as illustrated in Figure 5.

In the initialization phase, the electric motor controller and 2D laser scanner are parameterized, devices are started, and the whole system goes into the waiting state. In the waiting state, the hardware waits for a command from the control system. In the initialization phase of the 2D laser scanner, the FOV (field of view) and the angle resolution are set. During the initialization phase of the electric motor controller, the state machines are set to the right state and the periodic SYNC is set to CAN. In addition, the software is set to receive the rotation angle and speed data from the electric motor controller. Heartbeat receiving was also implemented.

The control system of the hardware implements two operating modes. "Drive to position" drives the 2D laser ▷

scanner to a certain angle according to the given parameters and "Sweep between angles" rotates the 2D laser scanner between certain angles with the given parameters. Both operating modes use the electric motor controller's internal operating mode "Profile Position Mode" [7]. Figure 6 presents the basic verification measurements of the electric motor drive.

The implementation of networks on the mobile machine is based on using CAN and Ethernet networks. The developed hardware uses both networks to transfer measurement data. Thus, the measurement data must be synchronized to the time domain in some way. One method for time synchronization is time stamping. Various devices can add time stamps to measurement data. For example, the 2D laser scanner and the electric motor controller can add time stamps to measured data when devices send measured data to the control system. Based on these time stamps, the measurement data can be time synchronized. This requires that the clocks of both devices are synchronized frequently enough to prevent time drifting.

Some devices don't provide time stamps for their measurement data or they are not usable for some reason. In this case, one must arrange time synchronization some other way. Thus, a deeper understanding of the behavior of networks and protocols is needed. Time stamping can be arranged in such a way that the receiver adds a time stamp to the measurement data. The receiver in this context is the middle level control. This time of arrival (TOA) is useful when there is not much of a data transfer lag and a small time synchronization error is allowed. With this hardware, time stamping is done in the following way: the 2D laser scanner sends the measurement data with a constant time interval, which is 25 Hz or 50 Hz. The measurement data is transferred from the 2D laser scanner to the middle level control via Ethernet. The middle level control then sends a SYNC request to the CAN network. The electric motor controller responds to this SYNC request with data that contains the rotation angle and the rotation speed.

According to the datasheet, the output of the 2D laser scanner is a real time output [6]. Thus, it is assumed to work in such a way that the 2D laser scanner measures the range data and sends the measured values to the middle level control through UDP with a low network latency right after measurement. The latency time in the data transfer in Ethernet networks is assumed to be negligible. We estimate that this latency time is less than one millisecond.

If the UDP packet sent by the 2D laser scanner arrives at the middle level control at time instant txPC, then the worst case scenario is that the SYNC request is sent to the CAN network from the middle level control at the time instant txPC + 1 ms. The electric motor controller responds to the SYNC request quite fast. Based on CAN analyzer measurement, the time that passes between the SYNC request and the response from the electric motor controller is less than 500 μs. In this context, we assume that the electric motor controller works as specified in the CANopen standard, which means the reaction to SYNC request comes within this measured time: sensors are read, the data is packed to a ▷

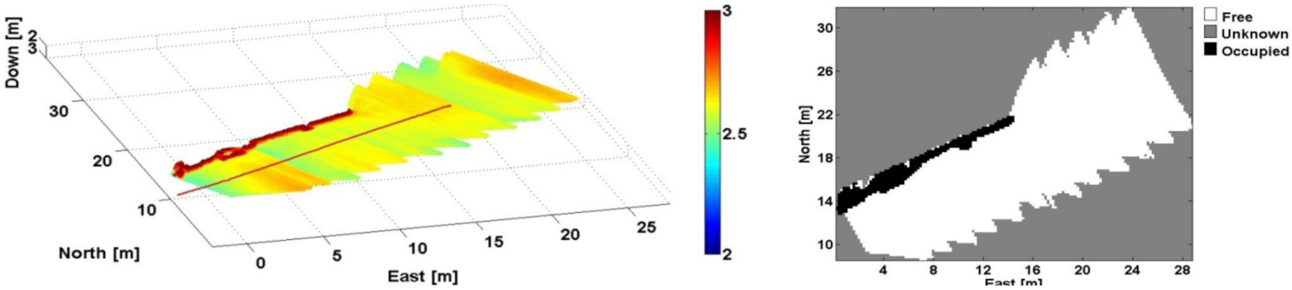


Figure 8: Point cloud created with low-cost Lidar installed on a GIM-machine and obstacle map created from the point cloud in question; the wall is presented in the point cloud with red and the wall is also recognized as an obstacle, shown black in obstacle map (Photo: TUT)

PDO, and the PDO is sent back to the middle level control. If the SYNC request is sent at the time instant tSYNC, then the response is available during the time instant tSYNC + 1 ms. When one takes all synchronization error sources into account, the total synchronization error is around 2 ms and it is caused by data transfer delay, program execution step size, and reading sensor to middle level control.

The internal operation of the 2D laser scanner also affects the synchronization error. The internal rotating mirror of the 2D laser scanner rotates with 25 Hz or 50 Hz so one cycle takes 40 ms or 20 ms. The UDP packet that contains the measurement data is sent at the time instance tS. This time instance takes place when the rotating mirror is in an area where the 2D laser scanner does not measure. This region is marked blue in Figure 7. One can't know this time instance exactly. The last measured point is measured at the time instance tM and this time instance differs fundamentally from the time instance tS. The time difference between these two time instances is around 5 ms to 10 ms if the measurement frequency is 50 Hz and FOV is 140 degrees. Figure 7 illustrates the region in red that the 2D laser scanner uses in this application. The time instance tM illustrated in Figure 7 is the place where the last measurement is done. When one takes all above-mentioned error sources into account, a maximum error of 15 ms time synchronization can exist between the last measured point and the rotation angle of the 2D laser scanner.

The time difference between the measured range values and the rotation angle causes a time synchronization problem that affects the laser map quality because the time synchronization problem causes a motion synchronization problem. In this context, a motion synchronization problem means that there is an error between the position of the rotating mirror and the rotation angle of the servo of the 2D laser scanner. Thus, the measured range values are not where the measured rotation angle expresses them to be. This error is related to the angular speed of the servo: a bigger angular speed means a larger error.

Operating with this hardware requires a time synchronization of the frame transformation. The coordination is needed between the rotating mirror of the 2D laser scanner and the axle of the electric motor drive. The related frame transformations are calculated in [8], where the motion synchronization is also taken into account by issuing an uncertainty for the frame transformation. The developed hardware requires calibration. Methods for calibration are not included in this paper but one must aware of this issue. The calibration method of our case is described in [8].

Mapping

The presented low-cost 3D Lidar has been used for mapping and path planning for autonomous mobile machines. The mapping method uses the servo angle and localization data to transform the laser range data into 3D point cloud in a global frame. Each point is also assigned with covariance data. The covariance for each point is calculated from the uncertainty data that is related to each coordinated transformation. A chain of transformations needs to be done in order to transform the range measurement of the laser scanner into a 3D point (x,y,z position) in the global frame. Each of these transformations has some uncertainty and this uncertainty propagates along the chain of transformations (visible in Figure 3).

The point cloud is then processed to the height map and further to the obstacle map. The mapping method and experiments are presented in [8]. An example point cloud and a related obstacle map are shown in Figure 8. The point cloud was created with the presented low-cost 3D Lidar system installed on a GIM-machine that is a modified multipurpose loader presented in Figure 8. The path driven with the GIM machine is presented as a red line in the point cloud data. One can see that the red area in the point cloud data is recognized as an occupied area in the obstacle map. The produced obstacle map can then be used for path planning as presented in [9].

Acknowledgement: This work was supported by the Academy of Finland through the General Intelligent Machines (GIM) project.

Authors

Antti Kolu, Mika Hyvönen, Petteri Multanen, Kalevi Huhtala, Kimmo Rajapolvi
Department of Intelligent Hydraulics and Automation
Tampere University of Technology
antti.kolu@tut.fi
www.tut.fi
References

ifm electronic

Look what we have for you!

An absolute must-have!

ecomat*mobile* BasicController*relay*: modular mini controller with powerful relay outputs with diagnostic capabilities for mobile applications. Direct connection of sensors and actuators without further external wiring, powerful CAN interfaces, programmable to IEC 61131-3 with CODESYS.
Can be extended with other components, such as the BasicDisplay. With E1 type approval of course.

www.ifm.com/gb/ecomatmobile_basic
Phone: +49 0800 16 16 16 4

Measuring linear positions

The Temposonics R-Series is a line of magnetostrictive linear position sensors designed for advanced motion control implementations. They come with a CANopen interface.

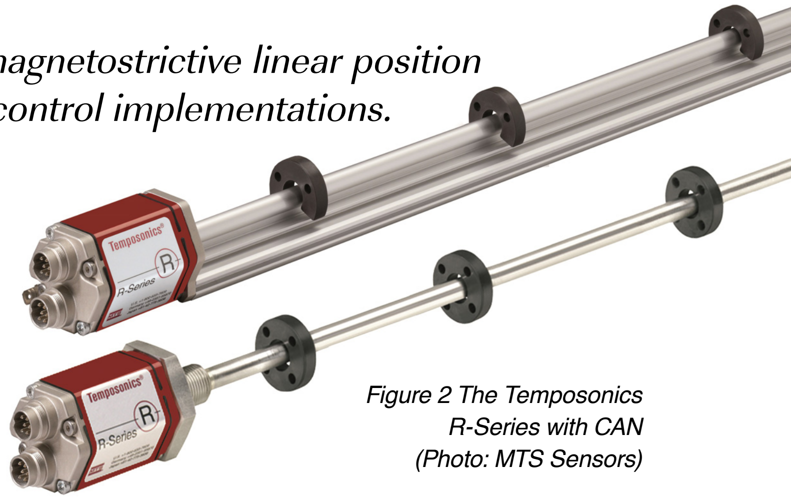


Figure 2 The Temposonics R-Series with CAN (Photo: MTS Sensors)

Extreme demands require extraordinary solutions. MTS Sensors responds to this with a range of measuring stroke options, simultaneous measurement of multiple magnets, smart electronic designs with built-in diagnostics, housing concepts, and a wide variety of controller interfaces. The Temposonics magnetostrictive technology and its designs provide reliability, position measurements, and long-term operation in harsh environments. The Temposonics R-Series is a line of magnetostrictive linear position sensors designed for advanced motion control implementations. Available in a rod version (RH), profile version (RP), with detached electronics (RD4), and with a flexible rod, the R-Series can be integrated into a range of applications. It features a double-shielded design, assuring immunity against EMI (electromagnetic interference), as well as modular construction.

The CANopen interface allows the simultaneous detection of four magnets using only one sensor with a stroke length up to 20 m: an advantage particularly in the print and paper industries. Further typical applications for the CANopen output are: renewable energy, testing machines, and metal working processing. For extremely harsh environments MTS Sensors offers the RS sensor with an IP69K protective housing. For situations where space-constrained mounting is a critical factor MTS proposes the Temposonics E-Series, which features a variety of slim designs. The EP2 sensor's position magnet for example, can travel along the entire flat housing profile without any limit. Due to the CANopen interface, their robust signal transmission, and exact position measurement they are suitable alternatives to reed-chain based sensors or limit switches in textile, plastic injection molding and packaging machines, as well as machines in the woodworking industry.

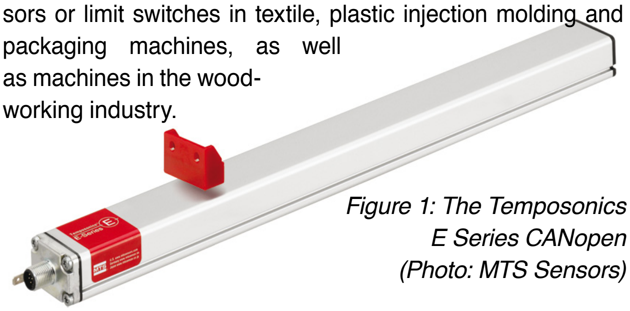


Figure 1: The Temposonics E Series CANopen (Photo: MTS Sensors)

Specifically designed for direct stroke measurement in mobile hydraulic applications the Temposonics MH-Series sensors can be fully sealed and embedded in a cylinder providing protection against the environment and EMI and enabling a long operating life. A MTS M12 connector system ensures protection to IP69K. The MH Safety model is SIL (Safety Integrity Level) certified according to IEC 61508, it has a Performance Level (PL) in accordance with ISO 13849-1 and meets the EN 954-1 standard.

CANopen in the wind industry

An important part of wind turbines is the hydraulic pitch control. Whenever the wind velocity and direction change, the pitch control adjusts the angle of the rotor blades by a few degrees to maximize the output for all wind speeds. When the wind velocity exceeds the maximum permissible generator output, the blades are adjusted away from the optimum position to reduce the aerodynamic efficiency and maintain constant rotation. Conversely, the blades are turned back into the wind whenever the wind speed drops again. Actively controlling the rotation speed not only enables peak efficiency but also reduces the stress on the rotor, the tower and the foundation for increased safety and longevity. Temposonics sensors with CANopen output are used for the position feedback in the closed loop control for the pitch of the rotor blades: they measure position with high accuracy even under harsh environment that wind turbines operate in with tremendous aerodynamic forces and continuously changing wind conditions. To ensure highest safety during maintenance, locking cylinders keep the rotor blades of wind turbine stationary and positioned precisely. The extending and retracting movement of the locking bolt makes the wind turbine lock and unlock. The detection of locking / unlocking position can also be monitored with MTS position sensors. They provide operation and functionality in the difficult maintenance situations due to rotor height or the challenging climatic conditions of offshore turbines.

Author

André Beste
MTS Sensors
info.de@mtssensors.com
www.mtssensors.com


FAULHABER
FAULHABER Motion Control
It's not a trick.
It's engineering.
FAULHABER
FAULHABER
NEW
WE CREATE MOTION
FAULHABER Motion Control Systems Series MCS 3242 / 3268 ... BX4
It is hard to believe how much intelligent and high-performance drive technology we have conjured up in the compact FAULHABER Motion Control Systems. Their robust construction in accordance with IP 54 meets even the most severe industrial requirements. And, equipped with RS232, EtherCAT and CANopen interfaces, they secure future connections. Ready for networked industry? With FAULHABER, it's not a trick.
www.faulhaber.com/mc/en
Easy commissioning with the new Motion Manager 6

(Photo: TWK)

Each vibration counts

Vibration monitoring is becoming increasingly important in machines and systems. The increasing demands are necessitating corresponding safety components. TWK offers a range of certified SIL devices.

Is there such a thing as good and bad acceleration? You may perhaps think that this is a strange question, but it is entirely justified. Shaking may well be a good thing for a cocktail, but it isn't so good in a system or machine. The task is therefore to ascertain whether acceleration caused by oscillations and vibrations can still be tolerated or whether it is already so high that the application is being damaged and has to be stopped.

TWK has developed the SIL2 vibration sensor NVA with CANopen Safety interface and switching relay for precisely this purpose. It is able to measure dynamic acceleration in a frequency range from 0,1 Hz to 60 Hz. Band filters can be used to subdivide the frequency range into sub-areas so that, for instance, low frequencies of less than 5 Hz can be analyzed more precisely and higher frequencies do not act as a disturbance, and vice versa. The CANopen Safety interface is implemented according to the following profiles: CiA 301 version 4.2 (application layer) and EN 50325-5 (safety protocol) as well as CiA 401 version 3.1 (Profile for I/O devices – Part 1: Generic I/O modules). The NVA can be extensively parameterized using this interface. Specific procedures now have to be taken into consideration during safe communication and parameterization by means of this interface so that safety-relevant parameters are not simply changed and the safe functions are not therefore impaired.

This topic is easy to handle with CANopen Safety: let us start with parameterization. Special CANopen objects are used to write new values which are desired by the customer to the changeable parameters; for instance, objects 320x (x=1 to 6) are used to define all acceleration limit values. However, the following is now important: so-called checksums additionally have to be transferred for each parameter change. The CANopen Standard profile differs from the CANopen Safety profile in this regard. Of course, these checksums have to be calculated in advance by the user for each parameter to be changed so that he can then transfer them to the sensor. This calculation can be carried out using a TWK program tool. The parameters with which the NVA115 is to operate – the above mentioned limit values, for instance – are entered in this tool, and the checksum is displayed in the result window.

However, this calculation can also be carried out independently using the underlying calculation polynomial (CRC-CCITT: $x^{16}+x^{12}+x^5+1$), or a calculation tool to be programmed by the customer can be stored in the control system. Before any change can be made to the NVA, so-called valid flag 32FE must be deactivated. The valid flag is a type of sensor lock mechanism. It has to be deactivated by writing an '0' into this object.

Example: in the NVA115, the warning limit value LW of filter 3 should be set to 2048 digits. With a resolution of 4096 digits / g, this corresponds to a value of 0,5 g (~ 5 m/s²). The value 2048 (= 800_h in hexadecimal) is now written in object 3203_h sub-index 01_h. The changed value is entered in the CRC calculation tool's screen – all other values remain unchanged – and the new checksum is displayed: $D97F_h$). This value is then written to the respective object $32FF_h$ sub-index 03_h. Valid flag $32FE_h$ must then be activated again: 'A5' is transferred. If there is an error in the parameterization – i.e. an incorrect checksum – the NVA115 cannot be enabled or ▷

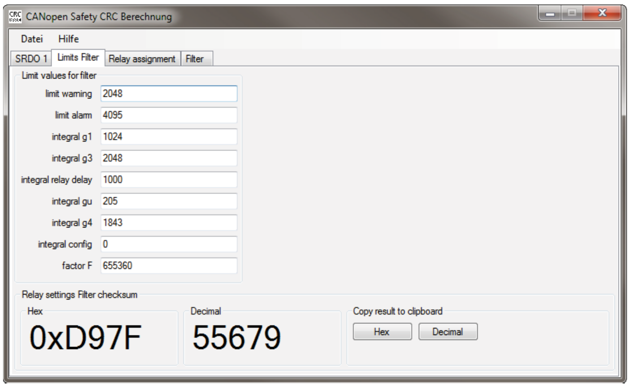


Figure 1: CANopen Safety CRC calculation screenshot (Photo: TWK)

started up using valid flag $32FE_h$. An error message is output if an attempt is made to write 'A5' to the object. If the value 'A5' is accepted, everything is OK, and the NVA can be switched to 'operational'. The NVA now operates with the new value.

The object structure of CANopen – in this case, CANopen Safety – permits this safe parameterization and significantly simplifies handling. No special device programs have to be used to carry out the changes. Unintentionally changing the parameters is virtually out of the question: to do this, the user "has to know what he is doing". Let us take a look at a specific example: Wind turbines generate electricity. However, the electricity does not simply come out of the socket or – as in this case – the wind turbine. Such a system is a highly complex and, in the meanwhile, extremely detailed construction which is an engineering work of art, and has to be protected from damage in order to minimize idle times and avoid costs. The oscillations and vibrations which occur during operation, primarily in the gondola and the mast, are important physical measured variables which have to be registered in order to protect the system. If the vibrations are excessively high, the entire system is affected. The acceleration forces which occur in the mast may lead to crack formation or even fractures. What are the possible causes? On one hand, they may be internal events. For instance, damage to the transmission or the bearings may lead to the occurrence of excessive main shaft vibrations. These vibrations lie in a frequency range from approx. 10 Hz to 50 Hz.

On the other hand, external influences may cause the system to vibrate. Amongst others, these influences include rotor blade icing or damage. These do not occur uniformly and therefore lead to rotor imbalance which can cause the entire system to vibrate. Or unfavorable wind conditions lead to excessive movements on the part of the gondola and therefore also the mast. The frequencies in this case typically lie between 0,2 Hz and 3 Hz. These vibrations have to be determined as part of vibration monitoring for a wind turbine in order to cause the control system to shut the system down if respective limit values are exceeded. This is where the NVA115 vibration sensor comes into play as a SIL2 Safety component. The measured acceleration value is constantly compared with limit values. If these are exceeded, internal safety relays are shut off. The two safety switching contacts, each of which in turn consists of two individual relays connected in series, are switched in the system's safety chain ▷

Figure 2: The NVA115 vibration sensor as a SIL2 Safety component (Photo: TWK)

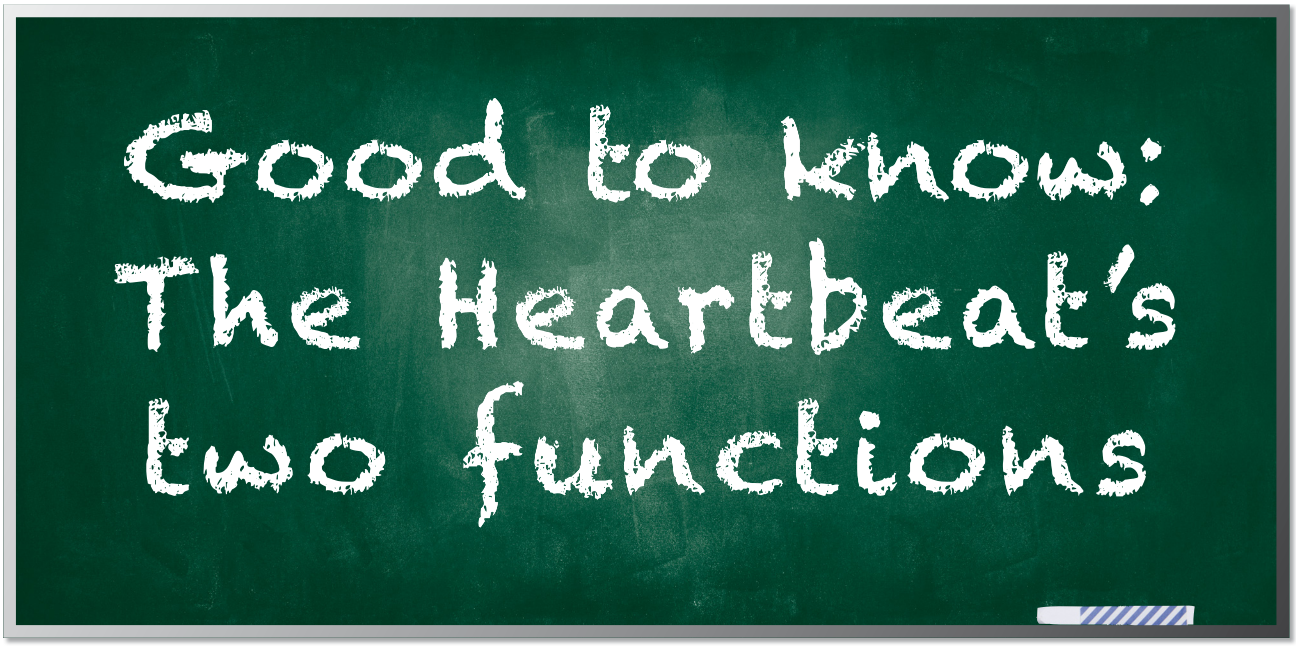
and, due to their series connection, ensure that the electrical circuit is safely disconnected - even under unfavorable conditions in which an individual relay would perhaps stick. One of the NVA115's special features is the fact that it can 'monitor' vibration behavior over a longer period of time. If a system's vibrations do not exceed a permissible amplitude value, everything is fine. If the value is briefly exceeded, however, the system does not have to be stopped immediately. Moderate and higher vibration values which occur temporarily are permissible if the system's vibration values subsequently decrease again. The NVA115 registers and evaluates precisely this behavior with its integral function: the safety chain is only interrupted when the system vibrates extensively for 'too long'. If suddenly occurring acceleration is excessive, however, the NVA115 reacts immediately (Safety-Shut-Off function SSO).

If the customer wishes, the vibration data are constantly transferred to the control system via the safe SRDO, which is 8 bytes long, in this case, the customer can see what is happening in their system in terms of vibrations. Of course, the above described safety shut-off function via the switching contacts is independent of this. Even when the NVA is switched to 'pre-operational', the NVA's safety system with the safety contacts is unreservedly operational. The SRDO is available twice: SRDO1 and SRDO2. The measured acceleration value is additionally transferred in bit-inverted form: safety first! After all, no incorrect data should be transferred unnoticed. This also applies from the sensor to the control system. As the NVA115 offers a range of different options for processing the measured acceleration value (RMS, PEAK, integral), it may be sensible to see all of these values for a specific filter; in the following, this is described for filter 3. The reason for this is that only the main value set via object 3223$_h$ sub-index 03$_h$ (output selection) is written to the SRDO. This may be the integral value, for instance. However, the other values which are also available, incl. the momentary value, can be read out at any time via the SDO data traffic (object 3283 for filter 3). All information is therefore accessible at all times if desired. However, the configurable safety switching contacts with the respective limit value comparison always react to a filter's output main value. Another special feature is that the NVA can be set to stop the system in a specific vibration phase. This is achieved using a shut-off delay T_A and the NVA115's ability to recognize the monitored vibration's zero-axis crossing. The positive zero-axis crossing is the starting point for the adjustable time T_A if the shut-off criterion, i.e. a limit value overshoot, was met beforehand. This time is set in object 3203$_h$ sub-index 05$_h$ for filter 3 - with subsequent checksum adaptation - and everything is ready. Of course, standard settings such as x or y axis assignment, momentary or RMS value are also possible with a number of adjustable parameters. Two analog signals are also optionally available for additional value output alongside CANopen Safety.

Finally, however, it is occasionally necessary to know whether the two switching contacts' relays are still operating reliably. It may be that they have not had to switch for a number of weeks or months. Despite robust relays and durable electronics, it makes sense to test them occasionally. To do this, it is possible to initialize a self-test lasting a few seconds via object 32FD$_h$. Depending on which value is transferred to this object, either switching contact 1 or 2 is tested, or both. However, the sensor remains in 'operational' status during the test, so that the control system can still see all of the current vibration data. The relay which is not being tested remains in safety mode and reacts to limit value overshoots. The self-test can be used to check whether the switching contacts really open, even if they are in the safety chain, as this can be deactivated for the duration of the test. So that the user can check the relay status from the point of view of the NVA115, they can read out this switching status with object 6300$_h$ sub-index 01$_h$. Incidentally, this can be done at any time, not only during the self-test. This is important so that the control system can detect which device in the safety chain has triggered, e.g. the NVA if the mast vibrations have become excessively high. With the NVA vibration sensor from TWK, an application is therefore always in safe hands for minimizing damage caused by interference acceleration and guaranteeing effective operation. All features, incl. the wide-ranging CANopen specialties, are precisely described in the company's detailed description, which is available on request. ◀

Author

Achim Albertini
TWK Elektronik
a.albertini@twk.de
www.twk.de



The Heartbeat protocol specified in the CiA 301 CANopen application layer and communication profile since version 4.0 has two purposes: detecting nodes that are not available and confirming NMT commands.

The periodically transmitted one-byte Heartbeat message indicates that a CANopen device is still alive and not bus-off or not powered. The Heartbeat consumer knows from the CAN-ID the node which node produced the Heartbeat message. The used CAN-ID has a base value of 700_h plus the node-ID. In the payload byte, the Heartbeat protocol indicates additionally its NMT status (pre-operational, operational, or stopped). This is so-to-say the confirmation on the received NMT command. The NMT master device should consume the Heartbeats of all connected NMT slave devices, to prove that the NMT command has been performed correctly.

The periodical transmission of the Heartbeat is configured by means of the Heartbeat producer time (Index 1017_h) given in milliseconds. The configured period is application-specific. The time should be shorter than the acceptable time to detect the absence of the device. If a device likes or needs to consume a dedicated Heartbeat, its Heartbeat consumer time array (Index 1016_h) has to be configured. The value is also given in milliseconds. The Heartbeat consumer time should be significantly larger than the Heartbeat producer time. At a first glance and as a rule of thumb, the doubled value is recommended. Of course, this also depends on the application requirements and the system designer may use other values.

The Heartbeat substitutes the old-fashioned Node/Life guarding mechanism. For new designs, CiA has recommended to implement the Heartbeat for more than 15 years. Each CANopen device should produce its Heartbeat. For devices without RPDOs and consumed EMCY messages it is not necessary to consume Heartbeats of other nodes. This is true for simple sensors, for example. Nevertheless, some motion control devices on the market don't provide Heartbeat consumer functionality. They can't detect a missing host controller by means of the Heartbeat. The CANopen conformance test tool does not detect this, because it is a matter of interoperability and not of conformity to the CANopen protocols.

The Heartbeat message provides also information about the current NMT status of the device. The NMT master device uses it as confirmation on its transmitted NMT command on the application level. For example: If the NMT master device has broadcasted the transit-to-operational command, it double-checks all received Heartbeats to see if all devices have transited to NMT operational state. Of course, some devices in the network may not be able to transit to the commanded NMT state within one Heartbeat producer period. This means that the NMT master device should be configured accordingly to tolerate a not matching status for a number of Heartbeat messages (in minimum one). Also, this depends highly on the application requirements and the performance of the selected NMT slave devices.

By the way, NMT master devices should also produce the Heartbeat message. Therefore, they need a node-ID assigned by the system designer. If you would like to diagnose the NMT master device regarding the configured Heartbeat producer and consumer times, it must provide an object dictionary. ◀

Author

Holger Zeltwanger
CAN Newsletter
headquarters@can-cia.org
www.can-cia.org

Interoperability of CAN FD transceivers

The C&S Group has developed an interoperability test system according to the CAN FD physical layer interoperability test specification. It has offered interoperability testing since November 2016.

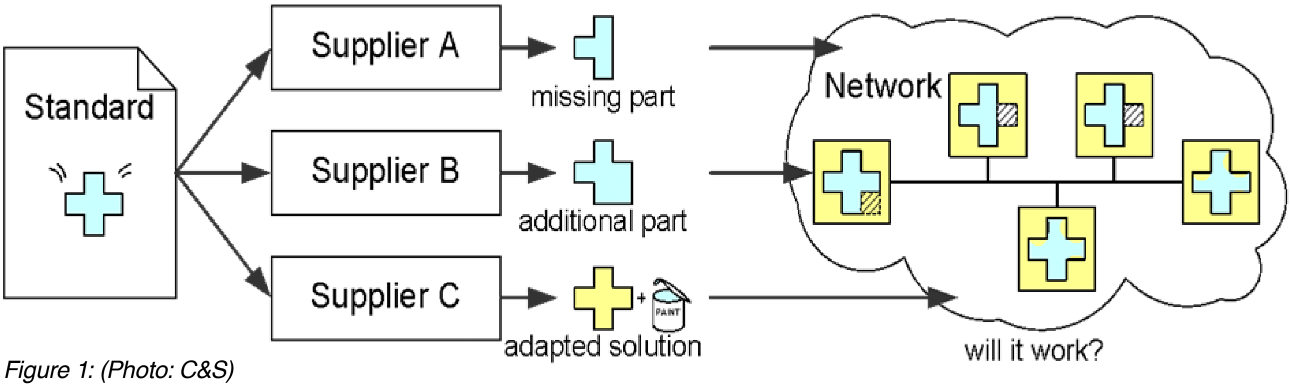


Figure 1: (Photo: C&S)

Since Bosch has released the first version of the CAN FD protocol specification, the first CAN FD transceivers, supporting communication in the CAN FD fast phase at higher data rates, are already available. Automotive manufacturers are currently working intensively on realizing the first CAN FD systems within their vehicle architectures. One building block to achieve this goal is to guarantee the correct inter-action of all components within the target distributed system – the interoperable behavior. After having addressed CAN FD conformance testing in the new international standards, ISO 16845-1 and ISO 16845-2, requirements from OEMs and silicon vendors were collected and aligned, and test cases have been drafted and specified to enable interoperability of CAN FD transceivers in a multi-vendor environment. The first release of the "Interoperability test specification for high speed CAN transceiver or equivalent devices" was published in June 2016. What are the differences between conformance testing (CT) and interoperability testing (IOPT)? The basic idea of conformance testing is testing to determine whether a product or system meets some specified standard that has been developed for efficiency or interoperability.

Fundamentals on conformance testing:

- To apply conformance testing, a specified standard must exist.
- Different implementations of a standard are existing or planned.
- The conformance test does not ensure the quality of the specified standard itself; it verifies the adherence of implementations of the standard to the standard.

Interoperability is a property referring to the ability of diverse products or systems to work together (to be able to interact, to communicate).

Fundamentals on interoperability testing:

- Interoperability is a property that is based on intended functional.
- Interoperability is relevant, if multiple entities shall interoperate.
- Specified standards shall describe interoperable products and systems, i.e. the intended functional behavior.
- Consequently, interoperability is a result of adherence of implementations to their specified standard.

It can be assumed that a solution of a single supplier, even if it would not adhere to the specified standard, is basically interoperable with other implementations of the same kind. If all share the same non-standardized behavior, they have a good chance to "interact" apparently correct. But if another implementation is introduced, a non-standardized behavior of an implementation might prevent the expected (specified) behavior in certain situations that are difficult to find in system-level tests and by try-outs. Therefore, the conformance test and the interoperability test need to be considered in case of multi-supplier solutions. If multiple suppliers create products or components based on the same specified standard, there is unfortunately a certain chance to create implementation containing deviations. Of course, each supplier has got own ideas on how to realize a product. Of course, all of them consider the specified standard. But, due to the fact that everybody has got a specific knowledge and a specific idea on the product, different suppliers may read the specified standard differently. Everybody knows that a message, a note or a text can be read by different people, resulting in different interpretations. This is even possible in very simple messages like "Buy some bread when you come home." Unfortunately, human language is very imprecise by nature: How much is "some bread": 200 g, 500 g, 1000 g? What kind of bread: white, grey, soft, with wheat, grains? What time will you come home: do you have to be there at a certain time?

The goal of all network and application designers is certainly interoperability and a correct application and system behavior. To achieve this goal, interoperability testing in addition to conformance testing is one further building block. Such interoperability tests are performed on standard system and typically care for the behavior of the application ▷

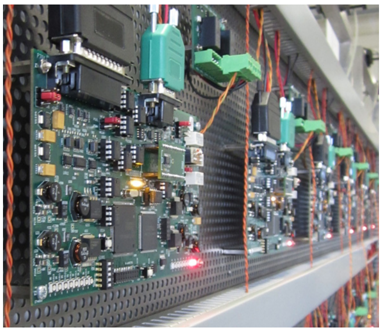


Figure 2 (Photo: C&S)

that is realized by a distributed system. The benefit is that such tests can be set up quite easy, by a simple mock-up. The implementations are tested for basic operations but also for stress conditions in terms of configuration, functionality, and error scenarios. In that way, it can be proven that each implementation adheres to the specified standard and that all nodes in a system can rely on the respective capabilities, ranges and limits given by the standard. The scope of the interoperability tests specification is the definition of test cases and test requirements to realize a test plan for the verification of high-speed transceivers or equivalent devices regarding their interoperability, even if provided by different manufacturers. The aim of the interoperability tests is to increase the probability of collaboration of CAN high-speed transceivers within a CAN system and to increase the confidence level in this regard. In contrast to conformance tests, the interoperability tests, which are defined within this test specification, are based on a predefined reference environment. Single device measurements are not the focus of the interoperability tests. The tests are performed within the reference environment, using predefined settings to ensure a high level of repeatability and comparability of the test results. The defined interoperability tests are focused on the transceivers or equivalent devices; for that reason, the additional devices, like common mode chokes or electrostatic discharge components, are not in use. The defined reference environments contain wire harness and passive components (resistances and capacitors) only.

Generally, the behavior of a transceiver or equivalent device can be represented by a state machine. The transitions from one state to another represent reactions to certain events e.g. mode change requests, bus failures, ground shifts (or their combinations). The behavior described this way is a dynamical sequential behavior. The defined interoperability tests verify the sequential behavior of the IUT in reference to the specified sequential behavior. The C&S Group has developed an interoperability test system according to the CAN FD physical layer interoperability test specification and offers interoperability testing since November 2016.

Author

Christoph Wosnitza
C&S Group
c.wosnitza@cs-group.de
www.cs-group.de

CANopen and the Raspberry Pi

Even in the industrial environment the Raspberry Pi is used as an entry-level computer. A variety of CAN connection options are available for the Raspberry Pi and there are different ways to support CANopen on it.

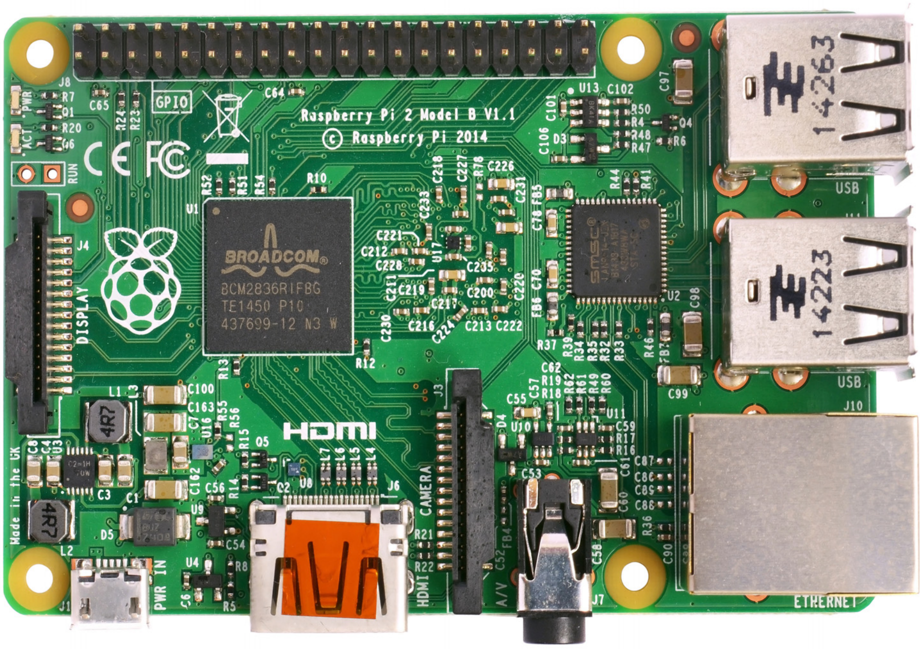


Figure 1: The Raspberry Pi 2 was used in a stamp machine (Photo: Multicherry, CC-BY-SA 4.0)

The Raspberry Pi is a series of small single-board computers developed by the Raspberry Pi Foundation. The basic goal of the foundation is to promote the teaching of computer science in schools and developing countries. After several previous generations, the Raspberry Pi 3 was released in February 2016. All models feature a Broadcom System on Chip (SoC), which includes an ARM-compatible CPU and an on-chip graphics processing unit. The onboard memory ranges from 256 MiB to 1 GiB RAM. Secure Digital (SD) cards are widely used to store the operating system and program memory in either the SDHC or Micro SDHC sizes. Most boards have various USB slots, HDMI, and composite video output. Lower level output is provided by a number of GPIO and the Pi 3 offers onboard Wi-Fi and Bluetooth support. The Raspberry Pi Foundation provides a Debian-based Linux for the board. Third-party providers offer various other Linux distributions or other operation systems. Unfortunately, none of the Raspberry Pi boards provide a CAN interface onboard, thus external CAN extension boards are required.

Industrial devices and clones

Customary Raspberry Pis are delivered as single-board computers without housing and without an industrial power supply. Because of their specific construction, derivatives are more appropriate for industrial use. One example for a derivative is the emPC-A/RPI by Janz Tec: the Raspberry Pi is boxed in an IP20 case and provides an industrial voltage supply of 24 V as well as improved EMC protection. Furthermore, the emPC-A/RPI provides additional interfaces. Especially the CAN interface including a transceiver and a DSUB-9-plug connector is highly attractive for CAN and CANopen applications. A back-end for CAN analysis software and a CANopen gateway are pre-installed by default. The back-end is the so-called Horch server, which is able to send and receive CAN messages and to transmit CAN telegrams via TCP/IP. Both the Horch server as well as the protocol are open source and hosted on Git Lab.

By using the Horch server, CAN applications can be created because of the access via the emPC-A/RPI and TCP/IP to the CAN network. Furthermore, different providers offer commercial CAN analysis tools using the Horch protocol. For example, the German software company Emtas offers their versatile CAN Interpreter and CANopen Device Explorer.

The Banana Pi is a Chinese clone of the original Raspberry Pi. Its developers have no connection to the Raspberry Pi Foundation. Nevertheless, the M1 and M1+ board devices provide an integrated CAN controller and CAN-RX and CAN-TX pins are available at the GPIO connectors. The integrated CAN controller provides a higher performance compared to a SPI-connected one. The CAN4Linux, another open-source Linux kernel device driver, has been ported to this target.

CAN connectivity

To achieve a CAN connection, a Raspberry Pi always needs an additional module often connected via SPI. One may develop these modules in-house or buy them as a ready-to-use component. These additional modules include a CAN controller and a CAN transceiver with an appropriate D-Sub 9 plug connector or screw terminal. ▷

Current versions of the Raspian Kernel provide a Socket CAN support for the MCP2515 CAN controller inclusively. The following steps are required to activate the Socket CAN modules:

- enable automatic loading of SPI in raspi-config
- modify /boot/config.txt to configure CAN controller on SPI
    - dtparam=spi=on
    - dtoverlay=mcp2515-can0-overlay, oscillator=16000000,interrupt=25
    - dtoverlay=spi-bcm2835-overlay
- activate CAN interface:
    - sudo /sbin/ip link set can0 up type can bitrate 250000

Provided that there is a support by a Linux device driver, one may alternatively use one of the USB ports to connect a CAN-to-USB interface. Some of the Raspberry Pis are equipped with up to four USB ports and enable users to connect more than one CAN interface. But for industrial demands, integrated solutions like Janz Tec's emPC-A/RPI are more suitable.

CANopen on Raspberry Pi

A CAN API – both Socket CAN or CAN4Linux – is the basis for CANopen protocol stacks or CANopen gateways. The software company Emtas provides a CANopen Master/Slave protocol stack running on the above-mentioned Raspberry Pi clones and variants. This protocol stack is available for Socket CAN and CAN4Linux and provides all CANopen features specified in CiA 301 and CiA 302. With this CANopen stack, various customer specific CANopen applications can be developed.

Another option is the use of the device as a CANopen-TCP/IP gateway according to CiA 309-3. Emtas provides source code variants of the 309-3 gateway that are adaptable to the customer's requirements as well as binary releases of the gateway which are pre-compiled for the emPC-A/RPI. Furthermore, a usable demo version is already pre-installed on these embedded systems. Even Modbus/TCP-CANopen gateways like CiA 309-2 run on this hardware. The CiA 309-3 specification defines an ASCII-based protocol where CANopen services are mapped into ASCII strings that can be exchanged through a TCP/IP socket.

The ASCII protocol for CANopen defines commands that are composed of tokens that are separated by white-spaces and finalized by CRLF characters. All commands sent to the gateway are confirmed and preceded with a sequence number that is enclosed in square brackets. The sequence number is an Unsigned32 number, which is sent back from the gateway with the answer. But these numbers are not used with event-triggered messages like PDO indications delivered to the TCP/IP client from the gateway. After the sequence number is sent, the command starts with an optional network-ID and the node-ID, which is addressed and followed by the specific command. All commands are defined in CiA 303-3 in Backus-Naur Form (BNF).

E.g. the definition for a SDO request:

"[" "]" [[net] node] r[ead]

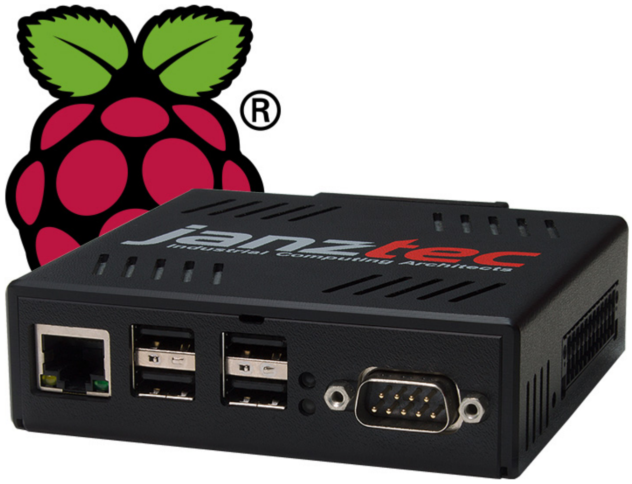


Figure 2: The emPC-A/RPI (Photo: Janz Tec)

An example for such a request:

[2232] 1 43 r 0x1000 0 u32

This means that the value of the object 1000_h sub-index 0_h shall be read from node 43 in net 1. The answer from the gateway might be the following:

[2232] 0x00010191

If a CANopen gateway – such as the one on the emPC-A/RPI – only supports a single CANopen network, the network number can be omitted. In addition to the exemplary SDO service, the CiA 309-3 protocol and the gateway also support other CANopen services like NMT, Heartbeat, Node Guarding, PDO, and LSS. Clients can connect these gateways via an RJ45 Ethernet jack. To connect the Pi3, even WLAN may be used in terms of a WLAN-to-CAN or -CANopen gateway. Once the task of the CiA SIG CANopen IoT, which was described in the previous edition of the CAN Newsletter, is completed, the CANopen gateway application by Emtas will also support URI queries and XML queries in addition to the ASCII protocol.

CANopen inside – the stamp machine

The control unit of a unique stamp machine is one notable example for the use of CANopen on Raspberry Pi. Developed by the engineers of Emtas, it underlines the wide range of CANopen applications. The control unit uses a Raspberry Pi 2 because the development was started before the availability of the emPC-A/RPI. Because of this, an external CAN board had to be used and SocketCAN was used as CAN API.

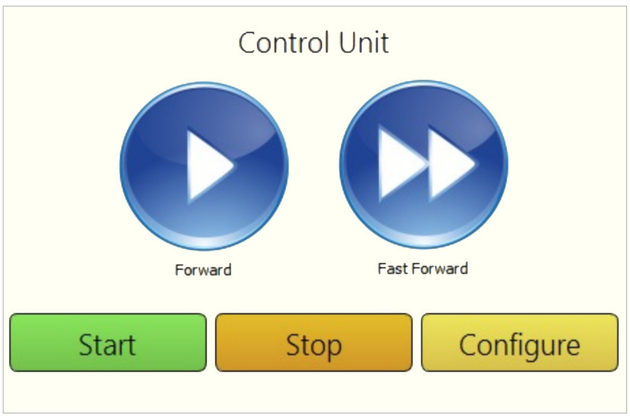
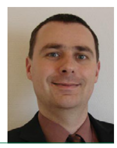


Figure 3 UI on touch display (Photo: Emtas)

In addition to the master – the Raspberry Pi – the network consists of five stepper drives with a CANopen interface that unfortunately does not support CiA 402. The task of the control unit is to control the movements of five motors moving single sheets of paper from a stack of papers to the stamp and to stamp it. The project required precise movements, but not necessarily fast reactions.

The 309-3 CANopen-TCP/IP gateway running in this Raspberry Pi had been used to configure and control CANopen devices. Because of the use of the CiA 309-3 protocol, it had been possible to run the control software on a desktop PC during the development. For future modifications of the machine, it will be possible to omit the touch-display at the machine and to control it from a PC via Ethernet. The control software had been developed using the Qt framework running on various platforms. The GUI and the control application were written with Qt. The UI of the application is displayed on a touch-display connected to the Raspberry Pi. The application communicates with the CANopen-TCP/IP gateway locally via TCP/IP sockets.

The Raspberry Pi and its clones or enhancements are suitable for various CAN and CANopen applications that range from CAN monitoring to the controlling of complete CANopen networks. Besides the aforementioned approaches that use protocol stacks or gateways, various IEC 61131 run time environments from various vendors are available for the Raspberry Pi too. Some of them also include CAN and CANopen support. The engineers at Emtas have gained a lot of experience with various embedded Linux controllers and are able to assist customers with CAN and CANopen projects based on the mentioned boards and devices.



Author

Torsten Gedenk
Emtas
ged@emtas.de
www.emtas.de

HY-TTC 32 - Compact Control Unit for Cost-Sensitive Applications, Smaller Machines and Implements

Flexibility and User Friendliness

- Extensive I/O set (30 I/Os with various options for configurability by software)
- Automatic baud-rate detection
- Programmable in C/C++ or CODESYS including support for CANopen® master
- 1 x CAN bus termination configurable via connector pin outputs

Safety

- Certified according to EN ISO 13849 PL c
- CANopen safety

Robustness and Availability

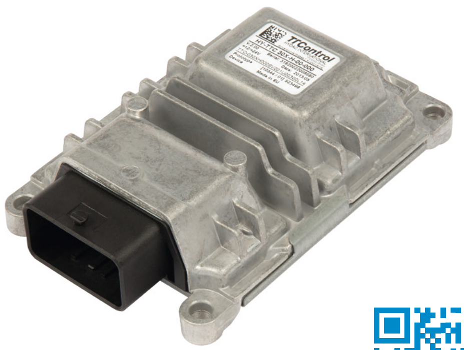
- Aluminum die-cast housing for extremely rough work conditions
- Maximum current up to 24 A

Connectivity

- 2 x CAN 125 kbit/s up to 1 MB/s
- CANopen conformity

Performance

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



www.ttcontrol.com

Safety Certified ECUs

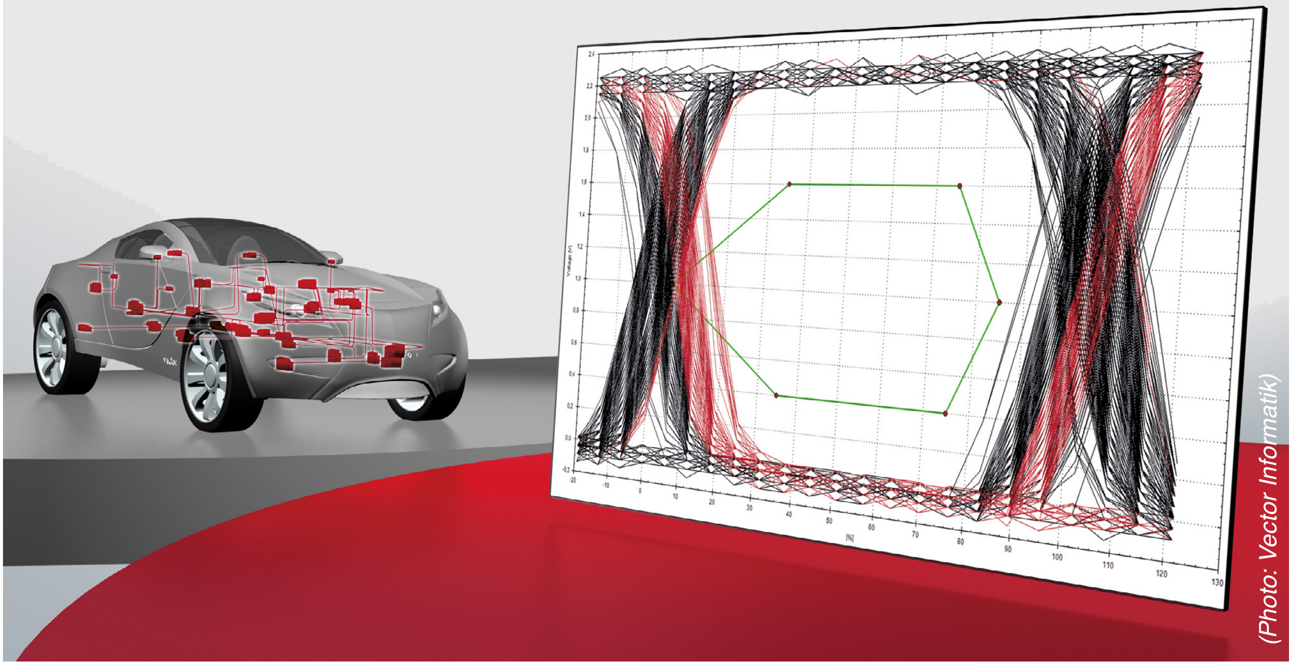
General Purpose ECUs

I/O Modules

Safe I/O Modules

Operator Interfaces

Eye diagram analysis for CAN FD



(Photo: Vector Informatik)

The eye diagram is an analysis method for evaluating the signal quality of transmission networks. Eye diagram analysis can help finding corrupting influences on CAN networks.

CAN networks can transmit information bit by bit from sender to partly distant receivers. The information transfer can, however, be easily corrupted by the network topology, cable length between the participants, line and terminating resistors as well as external electric influences. Both eye diagram analysis and serial bit mask analysis allow these influences to be identified and corrected early during the configuration phase of the CAN network.

Due to the higher and flexible bit-rate of the data phase, CAN FD is considerably more noise-sensitive than Classical CAN, which has a fixed bit-rate only for the entire frame. The transmission of CAN FD frames always begins with the lower bit-rate of the arbitration phase. A switch to the higher bit-rate of the data phase is made at the sampling point of the bit-rate switch (BRS) bit. At the sampling point of the cyclic redundancy check (CRC) delimiter bit, the bit-rate is switched back to the lower bit-rate. The shorter duration of the bits during the faster data phase has a negative effect on the signal quality, which can best be analyzed with an eye diagram. The signal quality is highly dependent on:

- Design and complexity of the bus topology e.g. line or star topology,
- Adversely selected cable routing and/or bus termination,
- Susceptibility of the system to higher transmission rates typical in vehicle networks,
- Cable properties, such as their impedances or shielding.

Creation of a classical eye diagram

During the network design, a configuration for the phase segments of the bit as well as their sampling points is created separately for the arbitration phase and the data phase. According to the CAN protocol, one bit is time-divided into four segments: a synchronization segment, a segment for compensating time delays, and two phase segments, which compensate the phase errors of the bit edges. The phase segments are often designated TSEG1 and TSEG2. The TSEG 1 summarizes the first phase segment and the compensation segment. Figure 1 shows the time division of a bit into the mentioned segments. The sampling point lies between the phase segments. TSEG2, on the other hand, corresponds to the phase segment which begins after the sampling point. Since the grid of the superimposed bits is displayed directly in the eye diagram, it is immediately apparent how well the superimposed bits fit into this grid.

The bit edges should ideally lie within the synchronization segment. The algorithm used here works like a real CAN controller. Due to the time delay of the bit edges, caused by the CAN transceiver delay and the jitter of the control units, CAN controllers must resynchronize at the transition from the recessive to the dominant bus level on the receiver side. This is a prerequisite for detecting the logic level of a bit at the set sampling point. With the adjustment of the phase segments, the robustness of the synchronization mechanism can be influenced. These ▷

settings are usually checked using an eye diagram, which allows the user to visualize whether his controller settings are practical and meaningful. Due to the fact that CAN FD has two different bit-rates, it is advisable to create separate eye diagrams for the arbitration phase and the data phase.

With the software tools CANoe and CANalyzer from Vector, the user is able to configure a CAN network and record voltage signals using the Option Scope. After measuring, the user can perform an eye diagram analysis, which analyzes all received frames bit by bit and superimposes them graphically in a fixed time window. In this example, the time window is a percentage of the bit duration (Figure 1). The displayed bit segments show the configured controller settings. Possible deviations in the individual bits can be quickly identified in this view. With a good controller setting, the rising edges of all bits lie in the synchronization segment. If the bit signals also reach their dominant and recessive voltage levels uniformly, i.e. without overshoot, a robust bus topology and a correctly selected bus termination can be assumed. The diagram has an "eye" due to the fact that all the bits on the x-axis are normalized to the theoretical bit-width (reciprocal of the bit-rate), while the voltage values of the bits are plotted on the y-axis. For the described case, the eye would be wide open (Figure 2). In the opposite case, the eye would be closed, which is an indication of errors in the network structure (Figure 3). The data phase is analyzed in Figures 2 and 3 with a sampling point set to 70 % and a data-rate of 2000 kbit/s.

Further refinement of analysis criteria

In order to narrow down possible sources of error, it is recommended to create an eye diagram from different aspects. For this purpose, various filter options are provided:

- Frame type, for example CAN or CAN FD,
- CAN channel number,
- Control unit name,
- Defined bit sequences.

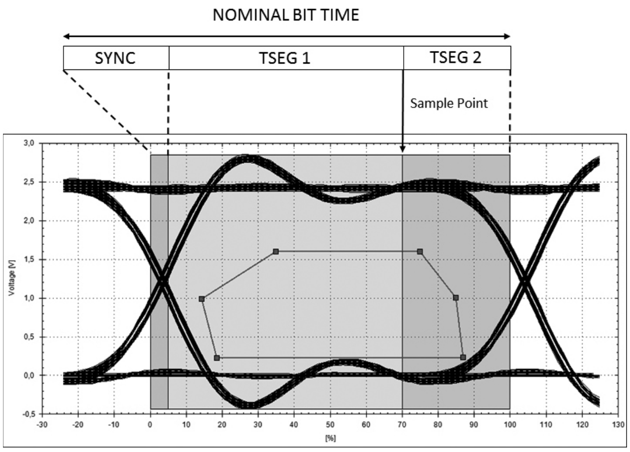


Figure 1: Classical eye diagram with graphical representation of the phase segments of a CAN controller (Photo: Vector Informatik)

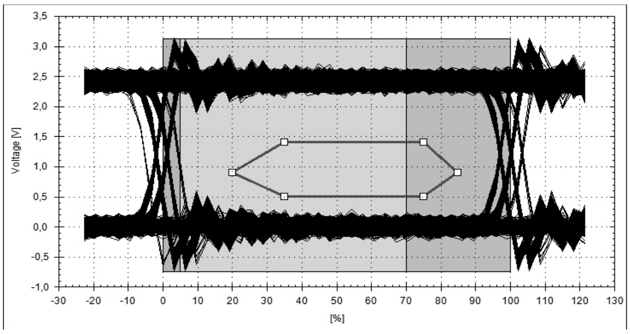


Figure 2: Open eye with bit mask – an indication of a good network design (Photo: Vector Informatik)

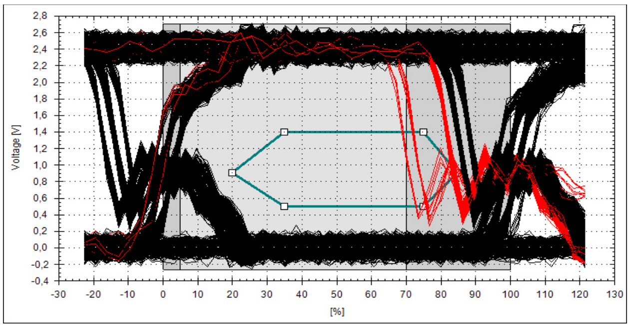


Figure 3: Closed eye with bit mask violation – an indication for problems in the network design (Photo: Vector Informatik)

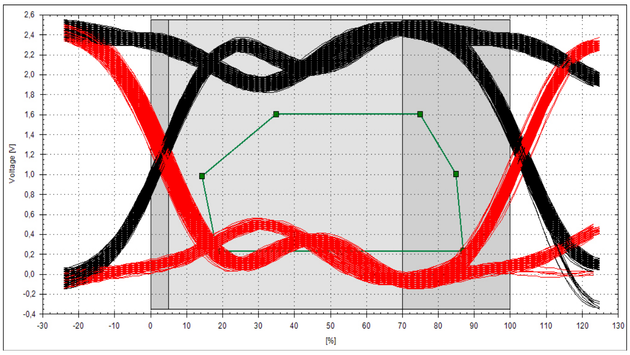


Figure 4: Eye diagram with bit mask violation (Photo: Vector Informatik)

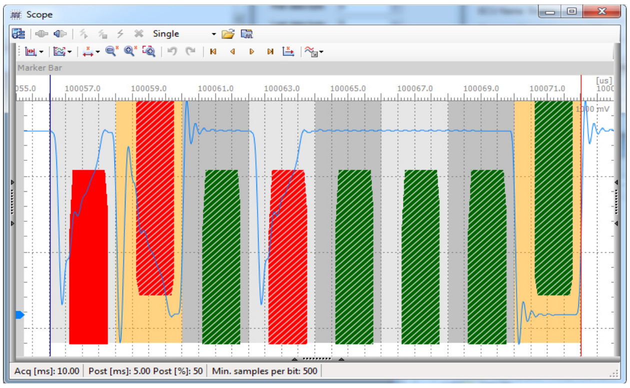


Figure 5: Serial bit mask analysis in CANoe/CANalyzer with a bit mask violation (Photo: Vector Informatik)

An alternative approach for detecting protocol errors of a control unit is the analysis of different bit sequences of a frame. With this method, a distinction can for example be made between the bits which are transmitted during the acknowledge phase by all control units and those of a particular control unit.

To further refine the criteria for evaluating the eye diagram, it is helpful to create a predefined theoretical eye in the form of a bit mask. For this purpose, the user generates a bit mask as a freely definable polygon, thus defining the "good" area, which may not be crossed by a voltage signal of any bit. It is useful to define a separate bit mask for the arbitration phase and for the data phase. Figure 4 shows an eye diagram with a bit mask, in which some bits violate the defined mask. In the background, the configured segments of the CAN controller are shown.

Serial bitmask analysis

So far, the classical eye diagram, which represents the overlapping of individual bits, has been considered. An alternative visualization is the serial bit analysis. In principle, both analysis procedures are performed identically. Only in the case of the serial bitmask analysis, the bits are displayed in the order sampled by an oscilloscope. Again, it is possible to define bit masks, which are displayed for every dominant and recessive bit. The advantage over the classic eye diagram is that bit errors are assigned directly bit by bit. It is also possible to analyze only part of the bit stream. The configuration possibilities already discussed for the eye diagram can also be applied to the serial bitmask analysis. Figure 5 shows all the bits of a defined analysis area. Each bit has a bit mask with the red mask indicating a violation in the first bit.

Automating the analysis process

Both analysis methods can be automated with the Vector product CANoe. To do this, the user must first define his test cases. For example, one test case is defined to analyze the data phase of all CAN FD frames and another test case for only the arbitration phase. For each test case, a specific bit mask can be used as a test criterion. If the bit mask is violated by the bit signal, the test case result will be negative. Each test case is automatically recorded, evaluated, and stored in a test report, so that the user can understand why a test case failed.

Conclusion

The methods described here for the analysis of a CAN/CAN FD network help to quickly identify and fix design faults or negative external influences. Eye diagram analysis with CANalyzer and CANoe is also available for other bus systems such as Flexray. With CANoe, the user also has the option to perform tests in a reproducible and automated manner via CAPL test sequences. Due to the high degree of automation and with suitable tools, such tests can be repeated with minimal effort.

Author

Mirko Donatzer
Vector Informatik
mirko.donatzer@vector.com
www.vector.com

Six-Channel CAN FD Interface

PCAN-USB X6

6-Channel CAN FD Interface for USB 2.0

The new PCAN-USB X6 is designed for the use in test benches with hardware-in-the-loop (HIL) simulations, in product line testing, and for working with multiple CAN or CAN FD busses.

- Adapter for USB 2.0 (compatible to USB 1.1 and USB 3.0)
- 6 High-speed CAN channels (ISO 11898-2)
- Complies with CAN specifications 2.0 A/B and FD
- CAN FD support for ISO and Non-ISO standards switchable
- CAN FD bit rates for the data field up to **12 Mbit/s**
- CAN bit rates from 25 kbit/s up to 1 Mbit/s
- CAN bus connection via D-Sub, 9-pin (in accordance with CiA® 303-1)
- Time stamp resolution 1 µs
- CAN termination at the CAN connection can be activated through solder jumpers, separately for each CAN channel
- Galvanic isolation up to 300 V, separately for each CAN channel on request
- High-speed USB 2.0 downstream port
- Voltage supply from 8 to 30 V
- Extended operating temperature range from -40 to 85 °C
- Measurement of bus load including error frames and overload frames on the physical bus
- Induced error generation for incoming and outgoing CAN messages

Driver, Software, and Programming Interfaces

Every PC interface from PEAK-System is delivered with a wide range of drivers, software, and programming interfaces. The scope of supply includes:

- CAN FD interface drivers for Windows 10, 8.1, 7 and **Linux**
- **PCAN-View:** Windows software for monitoring CAN and CAN FD busses
- **PCAN-Basic API** for developing applications with CAN and CAN FD connection for Windows (32/64 bit)
- **PCAN-PassThru** for using applications that are based on Pass-Thru (SAE J2534) with interfaces from PEAK-System
- Programming interfaces for standardized protocols from the automotive sector like:
  - **PCAN-CCP API** for the communication with ECUs according to the CAN Calibration Protocol
  - **PCAN-XCP API** for communication with ECUs according to the Universal Measurement and Calibration Protocol (CAN FD support since version 2)
  - **PCAN-ISO-TP API** for the transfer of data packages according to ISO-TP (ISO 15765-2)
  - **PCAN-UDS API** for the communication with ECUs according to UDS (ISO 14229-1)
  - **PCAN-OBD-2 API** for vehicle diagnostics according to OBD-2 (ISO 15765-4)

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