

<< emotas - your embedded communication partner <<

In the fields of automation technology and LEV development emotas is one of the leading providers of communication technologies.

With its product portfolio emotas is specialized in CAN, CANopen (FD), J1939 (FD), EnergyBus and UDS. Besides the sale of own products its software engineers develop customized embedded software solutions. The range of services includes consulting, development, on-site commissioning as well as software maintenance. Furthermore emotas provides training courses about the communications protocols CANopen, CANopen FD, UDS, EnergyBus, J1939 (FD) as well as trainings based on individual required content.

<< technologies <<

emotas is active member of the CAN in Automation e.V. (CiA) and development partner of the EnergyBus e.V. The experienced team is regularly working in different groups and in the committees for a standardization of CiA and EnergyBus e.V. Basing on these facts, emotas' products always do correspond to the latest state of technology.

what emotas stands for

- // expert advice
- // excellent Service
- // high quality standard
- // Made in Germany

<< sectors <<

emotas products are being used by diverse national and international industries:

- automation
- medical technology
- LEV components
- battery and charging technology
- lifts *and more...*

CANopen

CANopen^{FD}

J1939

 EnergyBus

UDS

ISO-TP

<< partnership <<

One of our major concern is to be always a reliable partner for our customers. Only together with them we may be able to shape a successful future of our company. Customers all over the world gain from strong service, professional consulting and high quality products.

<< service <<

Direct phone contact to the developers of the customers' product or project. At any time.

table of content

▶ CAN	6
RawCAN Stack	6
ISO-TP Client Server	6
▶ CANopen (FD)	7
CANopen (FD) Introduction	7
CANopen Protocol Stack	8
CANopen FD Protocol Stack	10
CANopen DeviceDesigner	12
CANopen NetworkDesigner	14
CANopen DeviceExplorer	16
CANopen Process DataLinker	18
CANopen Bootloader	20
CAN UpdateManager	21
CANinterpreter	22
CANopen Gateway CiA309	24
CAN Multiprotocol Stack	25
CANopen & J1939 training dates	27
CAN Hardware	28
▶ J1939 (FD)	30
J1939 Protocol Stack	30
J1939 FD Protocol Stack	31
J1939 DeviceDesigner	32
J1939 CANinterpreter	33
J1939 Bootloader	34

product portfolio

▶ UDS	35
UDS Server Stack Addon	35
UDS Client Stack Addon	35
UDS Bootloader	36
▶ Energy Bus	37
EnergyBus Introduction	37
EnergyBus Framework	38
EnergyBus DeviceWizard	40
EnergyBus Tool	42
▶ Services	43
Embedded Software Development	43
CANopen Integration	44
EnergyBus Integration	45
Commissioning Support	46
Trainings	47
10 Reasons to decide for emotas	48
Partners	49
About emotas	50

<< RawCAN Stack <<

The RawCAN Stack is a powerful CAN stack in source code. The stack is developed in ANSI-C and it is MISRA-conform. CAN devices can be quickly and easily developed or extended with our CAN Software Stack. Larger data packets can be transmitted by using the integrated ISO-TP functionality.

Compared to the basic CAN routines often provided by micro controller manufacturers our RawCAN stack and our own CAN drivers provide reliability and additional functions such as:

- cyclic transmission of CAN messages
- monitoring of received message (timeout)
- inhibit times for event triggered messages
- flexible transmit queue with transmission from interrupt
- receive queue with variable length

<< RawCAN Stack Highlights <<

- ANSI-C compatible CAN source code stack
- MISRA-C-conform
- available for many CAN controller and CPU types
- comfortable user interface
- widely configurable and scalable

A flexible user interface provides functions to evaluate the received data and to use the CAN services in the network. To connect the RawCAN Stack to multiple CAN controllers and CPU types, a well-defined driver interface is used. Using this driver interface the CAN stack can also easily be adapted to new CAN controllers or CPU types. Also it is possible to substitute hardware platforms with only little effort. The RawCAN Stack can be used with various Realtime Operating Systems.

To save resources the CAN Stack is widely configurable and scalable. The settings for these features are supported by the graphical configuration tool, CAN DeviceDesigner, which also allows the creation of the signal database. As a consequence, changes can be realized fast and easy. Using the unique CAN DeviceDesigner valuable development time is saved.

<< ISO-TP <<

Both ISO-TP server and ISO-TP client functions are integrated in the RawCAN stack.

Multiple sessions can be used in parallel to transmit larger data packets of up to 4 KB.

system requirements

- ▶ ANSI-C compiler

scope of delivery/licensing

- ▶ CAN-Stack Source Code (ANSI-C compatible)
- ▶ ready to run example application
- ▶ user manual and reference manual (Doxygen)
- ▶ incl. 6 months (Project License) or 24 months (Site License) support by e-mail or phone
- ▶ Named-User-License (Project), Floating License with dongle, Network License via license server for CAN DeviceDesigner
- ▶ optional maintenance agreement
- ▶ optional integration support

order information

- ▶ 5042-10 | RawCAN Pack
- ▶ 5042-10-ML | RawCAN Pack Multiline

<< Introduction to CANopen (FD) <<

CANopen (FD) is an industrial communication protocol based on CAN. Originally it stems from the automation technology and is internationally standardized (EN 50325-4). CANopen is nowadays being used within various fields of application: in machine controls, medical devices, rail- and off-road vehicles, E-bikes or building automation as well as in the field of conventional and renewable energy supply.

<< Overview <<

CANopen (FD) provides very flexible communication mechanisms defined in the CANopen communication profile CiA301 (CiA1301). Besides the CiA301 other profiles for applications, interfaces and devices are being developed and maintained by the CAN in Automation e.V. and its members under constant consideration of latest requirements. emotas is actively engaged in the further development of this and other communication profiles. By the experience subsequently gained from this work we are continuously developing our CANopen products and adapt it to customer's requirements.



<< CANopen DeviceExplorer <<

The CANopen DeviceExplorer is a versatile tool by emotas for development, testing, diagnosis and service tasks. This tool provides CANopen Master functions enabling the inspection and configuration of CANopen devices. The information about the respective CANopen device is being taken from the electronic data sheet of the device or may be extracted directly from the device. By means of the standardized DCF files (Device configuration files) device configurations may be saved and loaded. In project files even the management of whole CANopen networks is possible. Different Plugins extend the functional scope of the 'CDE'.

<< CANopen Protocol Stack <<

Developed in ANSI-C and MISRA-C-compliant, the emotas CANopen (FD) Protocol Stack is a software library providing all communication services according to the CANopen (FD) communication profile. Using the CANopen (FD) software you can quickly and easily develop CANopen (FD)-compliant devices or expand devices by CANopen (FD). The emotas protocol stack is available in the configuration levels Slave, Master/Slave and Manager.

<< CANopen DeviceDesigner <<

The CANopen DeviceDesigner is a very comfortable tool for designing CANopen (FD) devices quickly and easy. With a few clicks the user is able to assemble the object dictionary of the device and specifies their characteristics. Basing on this the CANopen DeviceDesigner generates the object dictionary, initialization functions, the electronic data sheet (EDS) and the device documentation. Additionally, the CANopen DeviceDesigner performs an optimal configuration of the CANopen (FD) Stack and the CAN driver.

<< CANopen Training and Consulting <<

The content of our offered trainings about CAN or CANopen (FD) are equally suitable for beginners and pros and may even be agreed individually. The topics range from CAN and CANopen (FD) basics to device and application profiles or specific issues related to your scope of application. The trainings addresses to development engineers and decision-makers who want to get acquainted with the technologies or intend to create themselves a basis for the work with our products. Even if you want to explore the technical capacities of CANopen - emotas is offering sound knowledge of CANopen for resolving complex questions by individual consultation.

<< CANopen Commissioning/Integration <<

emotas integrates the communication of your devices and supports you during commissioning for the best possible start of your project. All you need is your hardware, your development environment, a software engineer and the many years of know-how of our staff. The hardware is transferred during a joint appointment at your site or at our location in Merseburg.

<< CANopen Protocol Stack <<

The CANopen® Protocol Stack is a software library that provides all communication services according to the CANopen profile CiA 301. It is available in different variants, written in ANSI-C. Conformity to the standard is always tested with the recent CANopen Conformance Test of the CiA. The usage of the software helps to develop CANopen conform devices in a fast as well as comfortable way.

The flexible user interface allows an easy evaluation of received data and the comfortable usage of CANopen services in the network. Besides the standardized services in CiA301 and CiA302 there are further extension modules for different communication and application profiles available.

The defined driver interface provides a high degree of flexibility, because the adaption of the CANopen Protocol Stack to existing or new CAN controller or CPU types is easy to implement and therefore provides an easy exchange of the hardware platform. You may choose between different real-time operating systems (RTOS), Linux (SocketCAN, canlinux), Windows or QNX.

To save resources the emotas CANopen Protocol Stack is configurable and scalable. These features are implemented by the graphic configuration tool CANopen DeviceDesigner such as the creation of an object dictionary and EDS files, which are therefore always consistent. Changes may be easily updated.

- ▶ **CiA401** - device profile for digital and analogue IO modules
- ▶ **CiA402** - state machine for drives
- ▶ **CiA404** - device profile for sensors and regulators
- ▶ **CiA406** - device profile for encoder
- ▶ **CiA413** - device profile for commercial vehicles
- ▶ **CiA418** - device profile for batteries
- ▶ **CiA419** - device profile for chargers
- ▶ **CiA437** - application profile for photovoltaic components
- ▶ **CiA443** - application profile for SIIIS level 2 devices
- ▶ **CiA447** - application profile for electronic accessories in special vehicles (taximeter, flashing lights)
- ▶ **CiA454** - application profile for energy management systems such as in LEVs (EnergyBus)
- ▶ *and more...*

Flying Master

If CANopen Master were to fail, the Flying Master mechanism will detect it and the master functionality may be transferred to another device automatically.

Redundancy

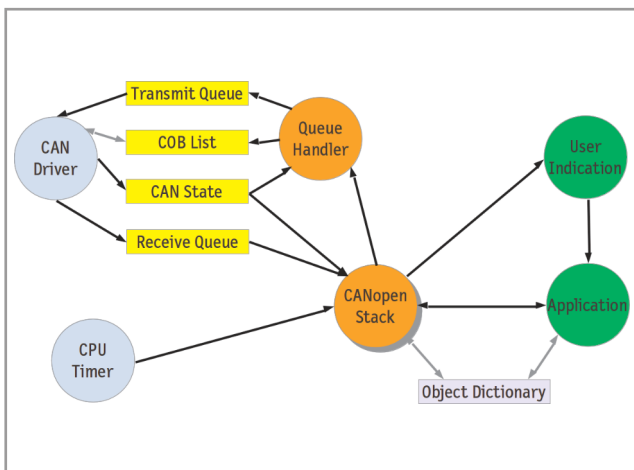
This means the usage of 2 CAN channels for transmitting process and configuration data. If one CAN channel were to fail, the second one may be still used. This extension is commonly used in maritime applications.

Safety

This extension provides support for the CANopen safety communication with SRDO (Safety Relevant Data Objects). Using this you are able to create applications according to SIL-3.

MultiLine

This extension enables the usage of the CANopen stack with several CAN interfaces. This is particularly interesting with regard to gateway applications.



<< CANopen extension modules <<

There are different extension modules for a multitude of CANopen device or application profiles. At present emotas is providing the following profiles:

<< CANopen Services <<

Services	Basic Slave	Master/Slave	Manager
SDO Server	2	128	128
SDO Client		128	128
SDO expedited/ segmented/block	●/●/-	●/●/●	●/●/●
PDO Producer	32	512	512
PDO Consumer	32	512	512
PDO Mapping	static	static/dynamic	static/dynamic
MPDO Dest.Mode		●	●
MPDO Src. Mode		●	●
SYNC Producer		●	●
SYNC Consumer	●	●	●
Time Producer		●	●
Time Consumer		●	●
Emergency Producer	●	●	●
Emergency Consumer		127	127
Guarding Master			●
Guarding Slave	●	●	●
Bootup Handling		●	●
Heartbeat Producer	●	●	●
Heartbeat Consumer	1	127	127
NMT Master		●	●
NMT Slave	●	●	●
Master Bootup CiA302-2			●
Flying Master CiA302-2		○	●
Configuration Manager CiA302-3			●
SRD CiA302-5		○	●
Redundancy CiA302-6		○	○
CANopen Router CiA302-7		○	○
LED CiA3032	●	●	●
Safety	○	○	○
LSS CiA305	●	●	●
ISO-TP		●	●
Multiline		○	○
C#-API-Wrapper Windows	○	○	○

● inclusive ○ optional

supported targets

The CANopen Protocol Stacks may be used with or without operating system. Numerous CAN controller and CPU types are being supported. For example: Bosch, Infineon, Intel, NXP, ST Microelectronics, Texas Instruments, Atmel, Renesas, Microchip.

The list of supported hardware platforms is being continually extended:

<https://www.emotas.de/en/info/controller>

If your required target is not available, the development of further driver packages is possible without any problem.

scope of delivery

- ▶ CANopen Protocol Sourcecode (ANSI-C; MISRA-compliant)
- ▶ CANopen driver
- ▶ CANopen DeviceDesigner
 - 1 Named User License (Project)
 - 1 Floating License (Site)
 or: Network License via license server
- ▶ ready to run example application
- ▶ (reference) manual
- ▶ optional maintenance agreement
- ▶ optional integration support

licensing

- ▶ Project License (Royalty free; incl. 6 months phone and e-mail support)
- ▶ Site License without runtime fees (Royalty free; incl. 24 months phone and e-mail support)

order information

- ▶ 5004-10 | CANopen Slave Pack
- ▶ 5004-11 | CANopen Master/Slave Pack
- ▶ 5004-12 | CANopen Manager Pack
- ▶ 2004-xx | CANopen driver
- ▶ 1004-4x | CANopen Profile
- ▶ 0140-10 | integration support

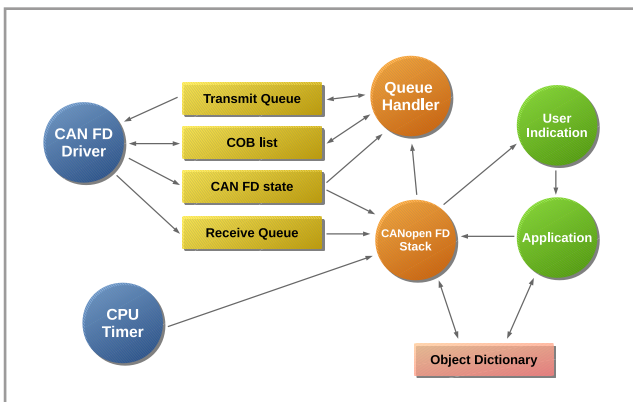
CANopen implementation examples for download: www.emotas.de/en/download

<< CANopen FD Protocol Stack <<

The CANopen® Protocol Stack is a software library that provides all communication services according to the CANopen profile CiA 1301. It is available in different variants, written in ANSI-C. Conformity to the standard is always tested with the recent CANopen Conformance Test of the CiA. The usage of the software helps to develop CANopen conform devices in a fast as well as comfortable way.

The flexible user interface allows an easy evaluation of received data and the comfortable usage of CANopen FD services in the network. Besides the standardized services in CiA1301 there are further extension modules for different communication and application profiles available.

The defined driver interface provides a high degree of flexibility, because the adaption of the CANopen FD Protocol Stack to existing or new CAN controller or CPU types is easy to implement and therefore provides an easy exchange of the hardware platform. You may choose between different real-time operating systems (RTOS), Linux (SocketCAN, canlinux), Windows or QNX.



Structure of the CANopen Stack

To save resources the emotas CANopen FD Protocol Stack is configurable and scalable. These features are implemented by the graphic configuration tool CANopen DeviceDesigner such as the creation of an object dictionary and EDS files, which are therefore always consistent. Changes may be easily updated.

The CANopen FD Protocol Stacks as well provide all services for classical CAN according to the CANopen profile CiA 301, so that compiled devices during its runtime may either use the classical CAN protocol or the CANopen FD protocol.

Please consider relevant restrictions when using classical CANopen. There are only 8 Byte PDO mapping data available instead of 64 and SDO must be used instead of USDO.

All Protocol Stacks are available as MultiLine versions for the use of multiple CAN interfaces.

<< CANopen extension modules <<

There are different extension modules for a multitude of CANopen device or application profiles. Concerning CANopen FD the specifications for these device or application profiles have not yet been completed. It will be provided as soon as it is finished.

At present emotas is providing the following profiles:

- ▶ CiA401 - device profile for digital and analogue IO modules
- ▶ CiA402 - state machine for drives
- ▶ CiA404 - device profile for sensors and regulators
- ▶ CiA406 - device profile for encoder
- ▶ CiA413 - device profile for commercial vehicles
- ▶ CiA418 - device profile for batteries
- ▶ CiA419 - device profile for chargers
- ▶ CiA422 - CleANopen, device profile for waste collection vehicles
- ▶ CiA437 - application profile for photovoltaic components
- ▶ CiA443 - application profile for SIIIS level 2 devices
- ▶ CiA447 - application profile for electronic accessories in special vehicles (taximeter, flashing lights)
- ▶ CiA454 - application profile for energy management systems such as in LEVs (EnergyBus)

.....
 The emotas CANopen FD Protocol Stack is also suitable for the development of standard CANopen devices.

<< CANopen services <<

Services	Basic Slave	Master/Slave	Manager
SDO Server*	2	128	128
SDO Client*		128	128
SDO expedited/segmented/block*	●/●/-	●/●/●	●/●/●
USDO Connections**	2	512	512
PDO Producer	32	512	512
PDO Consumer	32	512	512
PDO Mapping	static	static/dynamic	static/dynamic
MPDO Dest.Mode*		●	●
MPDO Src. Mode*		●	●
SYNC Producer		●	●
SYNC Consumer	●	●	●
Time Producer		●	●
Time Consumer		●	●
Emergency Producer	●	●	●
Emergency Consumer		127	127
Guarding Master*			●
Guarding Slave	●	●	●
Bootup Handling		●	●
Heartbeat Producer	●	●	●
Heartbeat Consumer	1	127	127
NMT Master		●	●
NMT Slave	●	●	●
Master Bootup CiA302-2*			●
Flying Master CiA302-2*		○	●
Configuration Manager CiA302-3*			●
SRD CiA302-5*		○	●
Redundancy CiA302-6*		○	○
CANopen Router CiA302-7*		○	○
LED CiA303	●	●	●
Safety*	○	○	○
LSS CiA305	●	●	●
Networking/Routing*	○	○	○
ISO-TP		●	●
Multiline		○	○
C#-API-Wrapper Windows	○	○	○

* Classic CAN Protocol Stack ** CAN FD Protocol Stack

● inclusive ○ optional

supported targets

The CANopen Protocol Stacks may be used with or without operating system. Numerous CAN controller and CPU types are being supported. For example: Bosch, Infineon, Intel, NXP, ST Microelectronics, Texas Instruments, Atmel, Renesas, Microchip. The list of supported hardware platforms is being continually extended:

<https://www.emotas.de/en/info/controller>

If your required target is not available, the development of further driver packages is possible without any problem.

scope of delivery

- ▶ CANopen Protocol Sourcecode (ANSI-C; MISRA-compliant)
- ▶ CANopen driver
- ▶ CANopen DeviceDesigner
 - 1 Named User License (Project)
 - 1 Floating License (Site)
 or: Network License via license server
- ▶ ready to run example application
- ▶ (reference) manual
- ▶ optional maintenance agreement
- ▶ optional integration support

licensing

- ▶ Site License without runtime fees (Royalty free; ; incl. 12 months phone and e-mail support)
- ▶ Project License (Royalty free; incl. 6 months phone and e-mail support)

order information

- ▶ 5004-10-FD | CANopen FD Slave Pack
- ▶ 5004-11-FD | CANopen FD Master/Slave Pack
- ▶ 5004-12-FD | CANopen FD Manager Pack
- ▶ 5004-15-FD-ML | CANopen FD Slave Pack MultiLine
- ▶ 5004-16-FD-ML | CANopen FD Master/Slave Pack MultiLine
- ▶ 5004-17-ML | CANopen Manager Pack Multiline
- ▶ 1004-4x-FD | CANopen FD profiles
- ▶ 2004-xx-FD | CANopen FD driver

CANopen implementation examples for download: www.emotas.de/en/download

<< CANopen DeviceDesigner <<

The CANopen DeviceDesigner is an easy-to-use tool for fast and cost-saving development of CANopen devices. With a few mouse clicks the object dictionary of the device can be created on the basis of predefined profiles. The CANopen DeviceDesigner creates the object dictionary and initialisation functions in C, the electronic data sheet in EDS and XDD format and a device documentation. Additionally the CANopen DeviceDesigner configures the CANopen (FD) stack and CANopen (FD) driver under consideration of the device characteristics.

The CANopen DeviceDesigner directs the user through the configuration of the hardware being used (microcontroller). During this process only a few characteristics have to be chosen: CPU type, CAN controller type and some timing parameters such as the frequency of the microcontroller.

According to the profiles and input, the object dictionary is created as C source code file. This can be integrated into the application as interface for the protocol stack. It is possible – when configured accordingly – to access the objects of the object dictionary directly as C variables. Alternatively there are access functions for accessing objects via index and subindex. Even a further processing of existing application variables is possible.

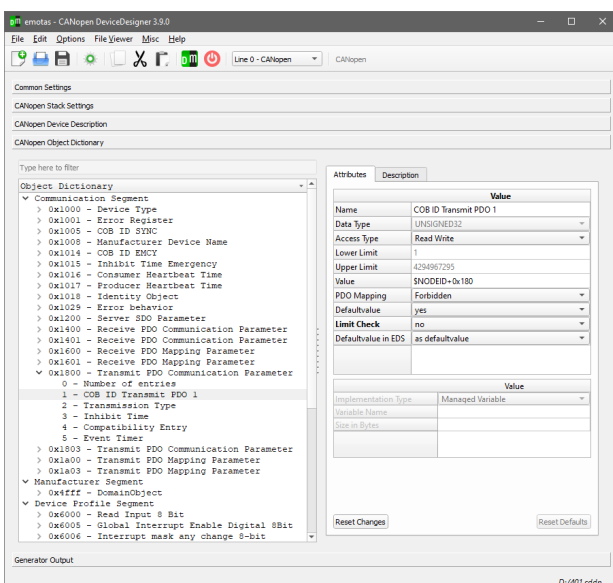
<< device profiles <<

For the communication profiles of the CiA e.V., there are databases for the CANopen DeviceDesigner available. A profile database includes templates for all objects of a CiA profile with all the default properties and a description to every object. It is possible to import these objects into the CANopen DeviceDesigner, duplicate and modify them. The usage of predefined databases shortens the developing time and eliminates the possibility of errors.

Optimized connection between application and stack by CANopen DeviceDesigner

At present emotas is providing the following profiles for the CANopen DeviceDesigner:

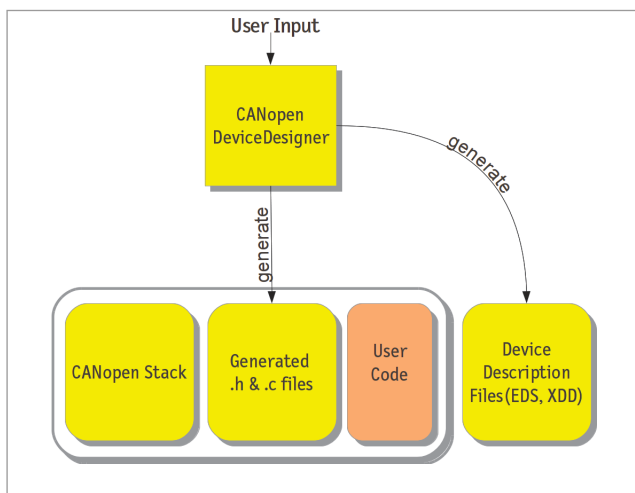
- ▶ CiA 301 - CANopen application layer/ communication profile
- ▶ CiA 302 - CANopen additional application layer functions
- ▶ CiA 401 - Generic I/O Devices
- ▶ CiA 402 - Drives and motion control
- ▶ CiA 404 - Measuring devices/closed-loop controllers
- ▶ CiA 418 - Battery modules
- ▶ CiA 419 - Battery Charger
- ▶ CiA 437 - Grid-based photovoltaic
- ▶ CiA 443 - SIIIS Level 2 devices
- ▶ CiA 447 - Special purpose car add-on Devices
- ▶ CiA 454 - Energy Management/EnergyBus



<< program output <<

From the user input the .h configuration files, the .c application code and the necessary device description files are generated. In the generating process the CANopen DeviceDesigner validates the input and generates helpful warnings. These generated configuration files configurate the CANopen Stack and activate only the particularly needed stack functionalities by setting compiler switches. The generated .c files contain the CANopen object dictionary, the PDO mappings and the code for the interface of the application. Hence these facts the emotas CANopen (FD) stack is highly optimized for the application and with regard to the memory requirement of RAM and Flash optimized for the respective application.

<< design flow <<



<< generated data <<

- object dictionary
- CANopen Stack configuration
- CANopen Stack initialization
- Electronic Datasheet (EDS according to CiA 306)
- XML Device Description (XDD according to CiA 311)
- HTML documentation of object directory
- list of objects in CSV format for import to spreadsheet software
- object description in txt format

system requirements

- ▶ Windows 11 or later
- ▶ diverse Linux variants tested with (K)Ubuntu
- ▶ emotas CANopen Protocol Stack

scope of delivery

- ▶ software download
- ▶ PDF manual
- ▶ profile databases for different CiA profiles

licensing

- ▶ Named User License/Node Locked License (with Project License inclusive to the stack)
- ▶ Floating License with dongle (with Site License inclusive to the stack)
- ▶ Network License via license server
- ▶ Corporate License on request

order information

- ▶ 1006-10/11 | CANopen DeviceDesigner Windows/Linux
- ▶ 1006-10-D/11-D | CANopen DeviceDesigner Floating License Windows/Linux

Evaluation versions of the tool available upon request.

<< CANopen NetworkDesigner <<

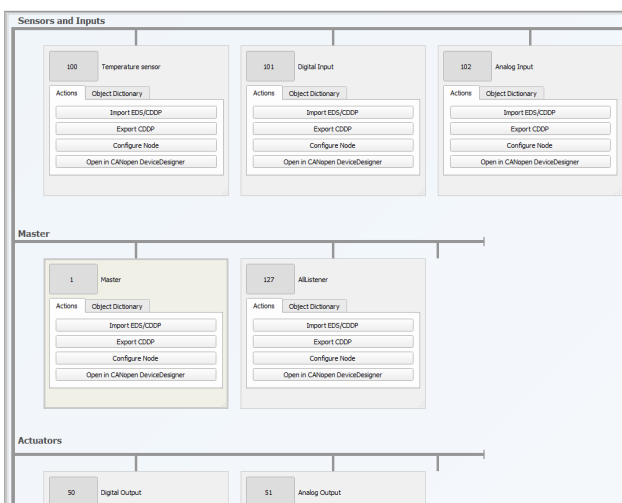
With the CANopen NetworkDesigner by emotas users may design a complete CANopen network including all devices and communication relationships. The source code for all components is generated automatically by the tool.

<< device and network design <<

Devices with all their signals and communications (via PDO and MPDO) can be defined with the CANopen NetworkDesigner. In this way the tool creates the source code for all components within the network. In combination with the emotas CANopen Stack the CANopen NetworkDesigner guarantees a CANopen compliant realization of the design. Heartbeat producer and Heartbeat consumer relations and the connection of Emergency producer and Emergency consumer are also done automatically.

<< object linking <<

During the configuration objects of producer and consumer are clearly shown in a table. Detailed CANopen notification is not shown here. As an example, instead of e.g. "object 0x6041, Subindex 1" the name of the signal "Control Word" leads users to a more convenient handling. By a single click objects of producer and consumer can be linked. Invalid linkings are excluded already in the run-up, because only valid links corresponding in their CANopen data type will be displayed. The individual prioritization of links is possible.



<< PDO Linking <<

According to settings done by users the tool sets the PDO configuration with mapping and COB IDs of the CANopen devices. In addition the manual configuration of PDO parameters by experts is possible. Then the CANopen NetworkDesigner verifies the valid configuration.

.....
 CANopen NetworkDesigner – latest generation of
 network design

<< simulation <<

A simulation of CANopen devices on a PC is also integrated into the tool. Not yet available devices and process inputs such as digital inputs, temperature sensors or other can be simulated with values. Within the simulated application the same CANopen Stack is used as in later real applications. So for the simulation it is not necessary to learn any further scripting language. All algorithms can also be used in the real devices.

<< network analysis <<

Project data of the CANopen NetworkDesigner can be imported to the CANopen DeviceExplorer. Hereby, the DeviceExplorer is able to interpret all signals in PDOs, SDOs and MPDOs. Further messages are also shown. Filters for CANopen devices and CANopen services are integrated. Additionally selected process data can be visualised graphically.

requirements

- ▶ Windows 11 or later
- ▶ diverse Linux variants
(tested with (K)Ubuntu)

scope of delivery

- ▶ software download
- ▶ PDF manual

licensing

- ▶ Named User License
- ▶ Floating License with dongle
- ▶ Network License via license server
- ▶ Corporate License on request

order information

- ▶ 1027-10 | CANopen NetworkDesigner
Named User License Windows
- ▶ 1027-11 | CANopen NetworkDesigner
Named User License Linux
- ▶ 1027-10-D | CANopen NetworkDesigner
Floating License Windows
- ▶ 1027-11-D | CANopen NetworkDesigner
Floating License Linux

<< CANopen DeviceExplorer <<

The CANopen DeviceExplorer is a versatile tool for development, testing, diagnostics and service tasks. It provides CANopen master functionalities and allows the analysis and configuration of CANopen devices.

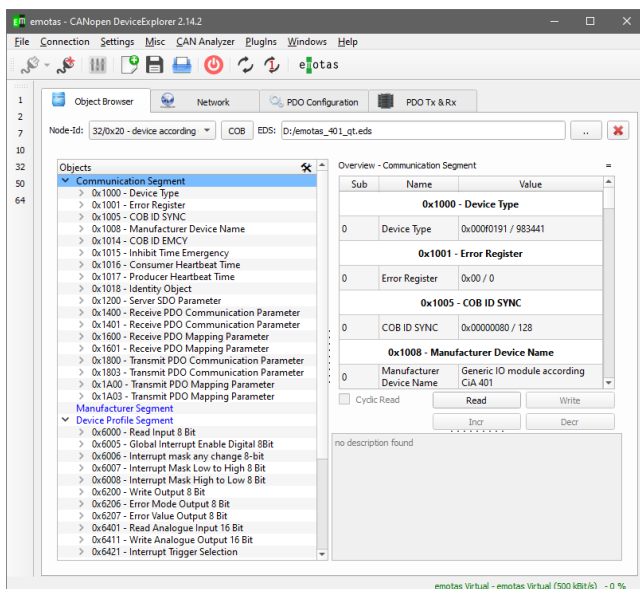
Information about each CANopen device is read from the electronic data sheet of the device(EDS, XDD) or they can be scanned directly from the device. Using standardized device configuration files (DCF) device configurations can be saved or imported. Additionally, data of entire CANopen networks can be stored in project files. The built-in scripting capability using QtScript allows users to create their own test and control applications with only little effort.

<< object dictionary <<

Objects of a device are presented in a clearly arranged object browser. According to the access rights, access to each object is provided for reading and writing. The CANopen DeviceExplorer supports all SDO services. Values of the objects can be displayed decimal, hexadecimal, binary or in ASCII format, depending on the data type. Large data like domain blocks can be read directly from or written to files. Objects being not described in the EDS file can be accessed as well.

<< network management <<

With the CANopen DeviceExplorer you are able to scan the network for already existing devices. Heartbeat and node guarding of the devices can be configured and monitored. Sending NMT commands to the devices is possible and the NMT state can also be monitored.



CANopen DeviceExplorer – an indispensable tool
 for your device development

<< PDO configuration <<

The tool supports the simple configuration of the PDO mapping by Drag&Drop. Additionally PDOs can be sent, received and visualized. For test purposes the SYNC message can be sent as one-shot message or cyclically with or without SYNC counter. The reception of timer-triggered PDOs can be monitored by the tool.

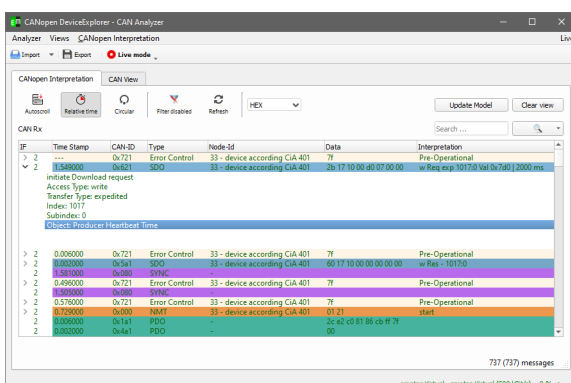
<< PlugIns – extension modules <<

The CANopen DeviceExplorer is available as **Basic** or **Full** version and can be extended by various optional PlugIns. The following PlugIns for the appropriate versions are currently available:

PlugIns/Variants	Basic	Full	separate
CANopen FD Support	●	●	
Object Browser (U)SDO Client	●	●	
NMT master	●	●	
PDO configuration	●	●	
PDO visualization	●	●	
Multi-Data Plotter	●	●	
DCF editor	●	●	
configuration download	●	●	
CAN analyzer	●	●	
CAN replayer	●	●	
CANopen Interpreter	●	●	
LSS Master		●	
CAN/CANopen Scripting		●	
SRDO Configuration			●
Process DataLinker			●
J1939 Interpreter			●
ISO-TP Interpreter		●	
EnergyBus Interpreter			●
CiA 402 Control		●	
Multi-Interface-Support		●	

more PlugIns for CANopen DeviceExplorer:

- ▶ 1007-20 | CDE PlugIn - Scripting
- ▶ 1007-21 | CDE PlugIn LSS
- ▶ EnergyBus PlugIns on request



system requirements

- ▶ Windows 11 or later
- ▶ Linux variants tested with (K)Ubuntu
- ▶ supported CAN Interface

CAN interfaces

Windows®

- ▶ IXXAT
- ▶ SYS TEC USB-CANmodul
- ▶ PEAK PCAN-USB
- ▶ Kvaser CAN-Interfaces
- ▶ EMS Wünsche CAN-Interfaces
- ▶ Vector CAN-Interfaces (XL-Driver)

Linux®

- ▶ can4linux
- ▶ SocketCAN

The Linux version supports all hardware interfaces with can4linux or SocketCAN.

scope of delivery

- ▶ CANopen DeviceExplorer for Download
- ▶ CANopen Interpreter PlugIn
- ▶ PDF manual

licensing

- ▶ Named User License for CANopen DeviceExplorer
- ▶ Floating License with dongle
- ▶ Network License via license server
- ▶ Corporate License on request

order information

- ▶ 5007-10/-11 | CANopen DeviceExplorer Basic Windows/Linux
- ▶ 5007-50/51 | CANopen DeviceExplorer Full Windows/Linux
- ▶ 5007-10-D/11-D | CANopen DeviceExplorer Basic Floating License Windows/Linux
- ▶ 5007-50-D/51-D | CANopen DeviceExplorer Full Floating License Windows/Linux

Test the CANopen DeviceExplorer.
We provide a demo on our website or on request at service@emotas.de

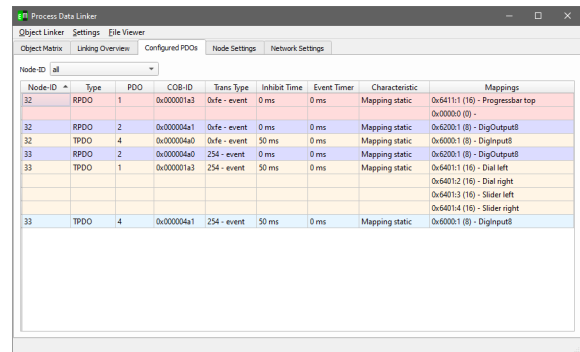
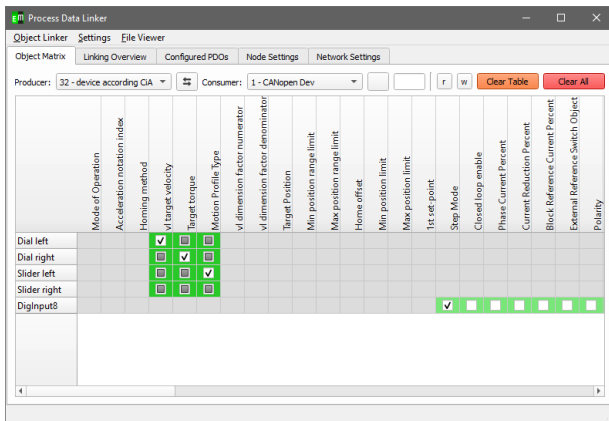
<< Process DataLinker <<

The Process DataLinker a powerful plugin for the CANopen DeviceExplorer for parameterization of communication links in CANopen networks. With a few clicks the user is able to define the links between the devices - the Process DataLinker automatically creates the PDO configuration. Furthermore no detailed CANopen knowledge is required and the PDO linking becomes more easier.

The CANopen DataLinker Plugin is an efficient expansion of the DeviceExplorer and is used to parameterize communication links in CANopen networks. You are able to define links between devices with a few clicks only. By this the CANopen DataLinker automatically creates the PDO configuration. No detailed knowledge of CANopen is required anymore and makes PDO linking notably easier.

<< PDO linking <<

According to settings done by users the tool sets the PDO configuration with mapping and COB IDs of the CANopen devices. Special cases like static mapping and fixed COB IDs are being considered as well. In addition the manual configuration of PDO parameters by experts is possible. Then the CANopen DataLinker verifies the valid configuration.



.....
CANopen DataLinker –
an indispensable tool for your CANopen network
configuration

<< object links <<

When configuring objects of producer and consumer are being presented in a table. Instead of the usual CANopen notation (0x6041, Subindex 1) an object name (i.e. Control Word) is being displayed and therefore leads to a clear presentation. Objects of producer and consumer may be linked by one click only. Invalid linkings are excluded already in the run-up, because only valid links corresponding in their CANopen data type will be displayed. Additionally it is possible to dermine priorities for specific linkings.

<< timer configuration <<

PDO specific timing parameters (i.e. Inhibit Time, Event Timer, SYNC timing) and heartbeat may be configured.

<< CANopen PLC Support <<

CANopen PLCs according to CiA405 are directly supported by the CANopen DataLinker. For each PLC network variables may be defined and linked with objects of other devices. In one project various PLC are possible. The definitions of variables according to IEC61131 are being created by the CANopen DataLinker directly.

```
File Viewer - /0/1007/software/cde/plcproject/plc02pdo_var_1.txt
(* IEC61131-3 Variable Definition File node 1 *)
VAR_EXTERNAL
  NV_U8_INPUT_1: BYTE; (* 0xa040:1 *)
  NV_U8_INPUT_2: BYTE; (* 0xa040:2 *)
  NV_I16_INPUT_1: INT; (* 0xa0c0:17 *)
  NV_I16_INPUT_2: INT; (* 0xa0c0:18 *)
  NV_U8_OUTPUT_1: BYTE; (* 0xa4c0:53 *)
  NV_U8_OUTPUT_2: BYTE; (* 0xa4c0:54 *)
  N3_AI0: INT; (* 0xa540:22 *)
  N3_AI4: INT; (* 0xa540:31 *)
END_VAR
(* generated by emtas CANopen Device Explorer - Process Data Linker *)
```

<< parameterizing of devices <<

Manufacturer specific objects and device-specific objects of devices can be configured with the CANopen DeviceExplorer. These settings are transferred into the CANopen DataLinker and the generated configuration files (DCF). Additionally all configuration files may be written into devices directly via CAN.

system requirements

- ▶ CANopen DeviceExplorer
- ▶ supported CAN Interface

CAN interfaces

Windows®

- ▶ IXXAT
- ▶ SYS TEC USB-CANmodul
- ▶ PEAK PCAN-USB
- ▶ Kvaser CAN-Interfaces
- ▶ EMS Wünsche CAN-Interfaces
- ▶ Vector CAN-Interfaces (XL-Driver)

Linux®

- ▶ can4linux
- ▶ SocketCAN

The Linux version supports all hardware interfaces with can4linux or SocketCAN.

scope of delivery

- ▶ CANopen Datalinker PlugIn Software for Download

licensing

- ▶ Named User License
- ▶ Network License via license server
- ▶ Corporate License on request

order information

- ▶ 5007-25 | CANopen DeviceExplorer incl. Process DataLinker PlugIn

<< CANopen Bootloader <<

The CANopen Bootloader by emotas is developed with special focus on resource saving characteristics. Only mandatory required CANopen services and objects according to CANopen profile CiA 302-3 for a bootloader are supported.

Only 16 KB of flash memory are sufficient for the full CANopen compliant bootloader implementation. Firmware can be updated with a standard CANopen Master or CANopen configuration tools. The CANopen Bootloader is available for different 16- and 32 bit microcontrollers. The portable driver interface allows the adaption to other target systems.

Optional extension modules like Layer Setting Service (LSS) for the configuration of Node Ids are available. Also manufacturer specific program code, as to sign firmware data, can be integrated fast and easily.

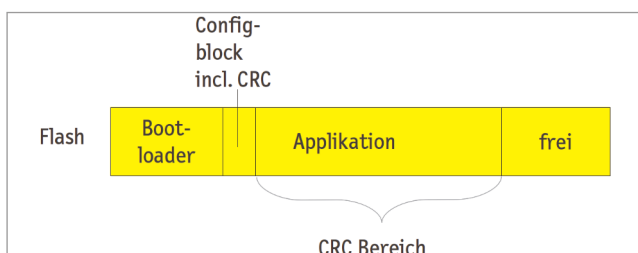
.....
low memory requirements - great benefit
.....

<< CANopen services <<

- ▶ NMT Slave - bootloader stays in pre-operational state
- ▶ Heartbeat - Heartbeat Producer Time defined at compile-time
- ▶ SDO Server - expedited and segmented SDO transfer SDO Abort - General Error
- ▶ Boot-Up - Send Boot-Up-message at the start of Bootloader

<< flash layout <<

The command line tool for the CRC-creation calculates the CRC from the binary file of the application. A configuration block with CRC is inserted at the beginning of the binary file. The Bootloader writes the binary file, including configuration block and CRC and application into the flash memory.



The delivery as source code allows to insert individual configuration blocks and to provide them for the application.

system requirements

- ▶ ANSI-C Compiler
- ▶ Microcontroller with In-Application Programming (IAP) support

scope of delivery

- ▶ Bootloader source code (ANSI-C-compliant)
- ▶ adjustment to CPU, first internal CAN controller and internal flash
- ▶ EDS file of the bootloader
- ▶ ready to run example application
- ▶ PDF manual
- ▶ command line tool for checksum generation for Windows and Linux
- ▶ incl. 1 Named User License (Project), Floating License (Site) or Network License via license server for CAN UpdateManager

licensing

- ▶ Project License (Royalty free; incl. 6 months phone and e-mail support)
- ▶ Site License without runtime fees (Royalty free; incl. 24 months phone and e-mail support)

order information

- ▶ 5001-10 | CANopen Bootloader generic

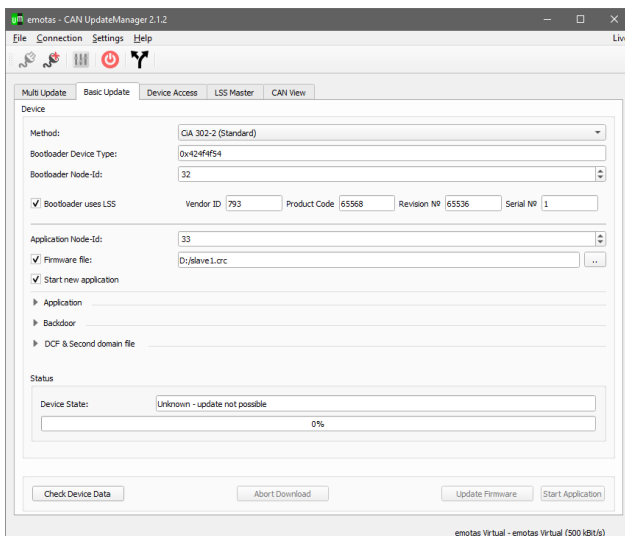
<< CAN UpdateManager <<

The CAN UpdateManager is a program for firmware downloads and was developed with focus on easy handling. The UpdateManager supports update mechanisms according to CiA 302-2 and also according to other application profiles including the access to password objects.

For developers of CANopen bootloaders the integrated CANopen Master with SDO client (expedited and segmented transfer) and NMT master allows direct access to the CANopen device. This makes it possible to successively validate that device and bootloader behave as expected. In addition the setting of parameters for CANopen devices is possible with the CAN UpdateManager. Communication of CAN UpdateManager and device can be recorded for diagnosis purposes.

firmware download and network update in one tool

By using the bootloader you are able to make a direct firmware download for a device or you can even bundle firmware files and configuration data into one update package. The site service technician only starts the tool with the update package to update all devices of a network automatically.



system requirements

- ▶ Windows 11 or later
- ▶ Linux variants tested with (K)Ubuntu
- ▶ supported CAN Interface

CAN interfaces

Windows®

- ▶ IXXAT
- ▶ SYS TEC USB-CANmodul
- ▶ PEAK PCAN-USB
- ▶ Kvaser CAN-Interfaces
- ▶ EMS Wünsche CAN-Interfaces
- ▶ Vector CAN-Interfaces (XL-Driver)

Linux®

- ▶ can4linux
- ▶ SocketCAN

The Linux version supports all hardware interfaces with can4linux or SocketCAN.

scope of delivery

- ▶ Software for download
- ▶ PDF manual

licensing

- ▶ Named User License (incl. a Project License)
- ▶ Floating License with dongle (incl. a Site License)
- ▶ Network License via license server
- ▶ Corporate Licenses for service technicians or Buyout versions on request

order information

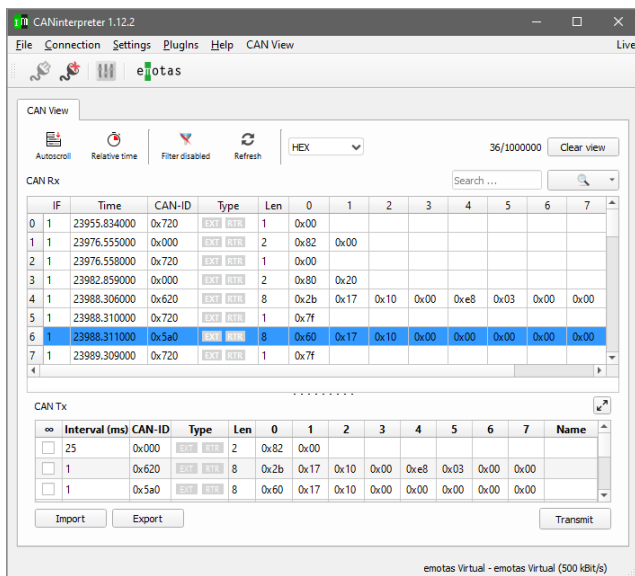
- ▶ 5002-10/11(-D) | CAN UpdateManager Basic
Windows/Linux
PlugIns: CANopen
- ▶ 5002-50/51(-D) | CAN UpdateManager Full
Windows/Linux
PlugIns: CANopen, J1939 (CAM11/CAM21), UDS

<< CANinterpreter <<

The CANinterpreter is a multi-purpose tool to monitor and record data traffic and for the analysis of CAN networks. Different plugins extend the scope of functions.

<< basic features <<

- online-monitoring of bus traffic in different modes (Trace/Object View)
- recording of CAN messages
- flexible filtering of CAN-IDs
- single or cyclic sending of CAN messages or sequences
- CAN FD ready when used with a suitable interface



<< PlugIns – extension modules <<

Highlight of the CANinterpreter is the modular structure that allows a wide range of functions by offering different optional PlugIns.

Creating PlugIns or manufacturer specific add-ons according to customers' requirements is also possible. Use cases for that could be e.g.: Use of the CANinterpreter as manufacturer specific service tool or as setup tool for devices and plants.

Scripting PlugIn

The Scripting-PlugIn is based on QtScript/ Javascript extended by CAN and CANopen specific commands. For example to send or receive CAN messages or different CANopen services. User interfaces tailored to customers' demands can be created with the graphical UI-Designer. Numerous examples belong to the standard delivery.

PlugIn EnergyBus visualization

This PlugIn has been developed in particular for the use with EnergyBus. It monitors the data traffic with EnergyBus (CiA 454) and displays the process data and conditions of all EnergyBus devices within the network.

PlugIn CANopen interpretation

This PlugIn enables the interpretation of CANopen messages on the basis of CANopen services. Instead of data bytes „40 10 17 00“ the PlugIn indicates that the Heartbeat Producer object (0x1017) should be read by SDO. Appropriate interpretations for all CANopen services are available. Data of the PDOs can be shown by using the mapping. Furthermore a filtering of messages by node-ID or CANopen service is possible.

PlugIn J1939

The J1939 PlugIn is an optional extension of the CANopen Device Explorer or the CANinterpreter and provides functions for interpreting and sending J1939 messages. Individual SPN values can be displayed graphically in charts.

PlugIn UDS

The UDS PlugIn is optionally available for the CANopen DeviceExplorer or the CANinterpreter and provides functions for interpreting and sending UDSONCAN messages.

system requirements

- ▶ Windows 11 or later
- ▶ Linux variants tested with (K)Ubuntu
- ▶ supported CAN Interface

CAN interfaces

Windows®:

- ▶ IXXAT
- ▶ SYS TEC USB-CANmodul
- ▶ PEAK PCAN-USB
- ▶ Kvaser CAN-Interfaces
- ▶ EMS Wünsche CAN-Interfaces
- ▶ Vector CAN-Interfaces (XL-Driver)

Linux®:

- ▶ can4linux
- ▶ SocketCAN

The Linux version supports all hardware interfaces with can4linux or SocketCAN.

scope of delivery

- ▶ CANinterpreter Software for download
- ▶ PDF manual

licensing

- ▶ Named User License
- ▶ Floating License with dongle
- ▶ Network License via license server

order information

- ▶ 5017-10/11 | CANinterpreter Windows/Linux
- ▶ 5017-10-D/11-D | CANinterpreter Floating License Windows/Linux
- ▶ 1007-20 | PlugIn - Scripting
- ▶ 1007-22 | PlugIn CANopen Interpreter
- ▶ 1007-27 | PlugIn J1939
- ▶ 1017-30 | PlugIn UDS
- ▶ EnergyBus PlugIns on request

<< CANopen Gateway CiA-309 <<

The universal CANopen Gateway software according to CiA 309-3 provides access down to the field level using the reliable and proven CANopen technology.

<< basic features <<

The CiA309-3-Gateway is a multipurpose TCP/IP CANopen gateway according to CiA309-3. Its binary version is available for Linux and may be used with any Linux providing a can4linux or SocketCAN interface. Furthermore the gateway is available as source code and may be ported to every platform provided that they have CAN interface and a TCP/IP Stack with BSD socket. The CANopen module of the gateway is the proven CANopen Master Stack of emotas.

<< CANopen services <<

Following CANopen services defined by the CiA 309-3 are currently supported by the gateway:

- SDO Client (data types from 8..32 Bit, strings and domains)
- PDO Consumer
- PDO Producer
- NMT commands
- Node Guarding Master
- Heartbeat Consumer
- Heartbeat Producer
- Emergency Consumer

More CiA309-3 commands for a configuration of the gateway are available.

Using the CANopen TCP gateway according to CiA309-3 CANopen devices may be easily integrated in TCP networks. Further applications going beyond the functional scope of the CiA309-3 specification may be developed with the source code version basing on the CANopen Master Stack.

system requirements

- ▶ Raspberry Pi
- ▶ Microsoft 7, 8,10 or diverse Linux variants tested with (K)Ubuntu)

scope of delivery

- ▶ CANopen TCP Gateway Software for download
- ▶ PDF manual

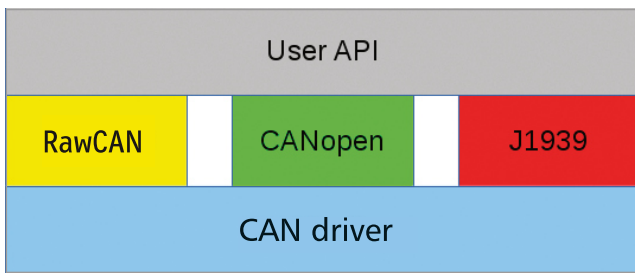
order information

- ▶ 1021-xx | CiA 309 CANopen TCP Gateway

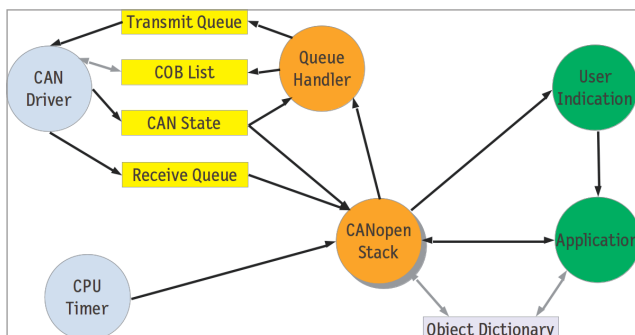
<< CAN Multi Protocol Stack <<

The CAN Multi Protocol Stack is a software library that provides services to implement CAN, CANopen and J1939 applications with one or more lines in one device. The included graphic configuration tool supports the fast and comfortable development.

The CAN Multi Protocol Stack is written in ANSI-C and therefore applicable on various systems. A flexible application programming interface provides functions to evaluate received data and to use CAN, CANopen and J1939 services in the network.



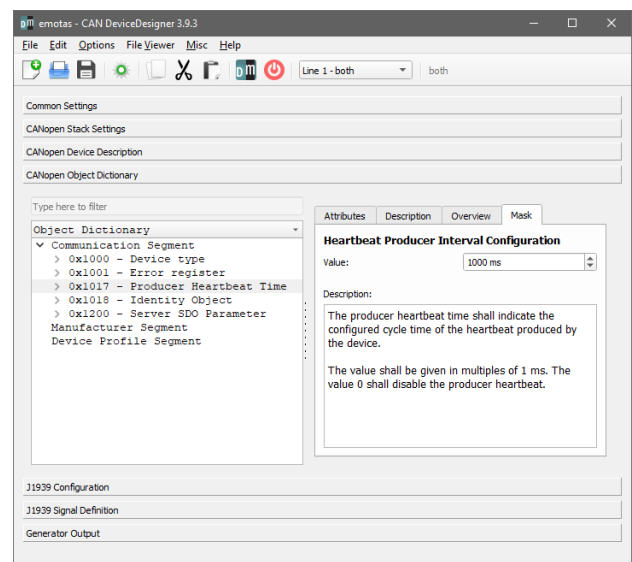
The implementation of CANopen provides every communication service according to CiA301 V4.2 and is available in various expansion stages. Compliance with the CANopen standard is always tested with the latest CANopen Conformance Test by CiA e.V. Extension modules for several communication and application profiles, besides those of the CiA 301 and 302 are available.



To connect the CAN Multi Protocol Stack to multiple CAN controllers and CPU types, a well-defined driver interface is used. Using this driver interface the stack can also easily be adapted to new CAN controllers or CPU types. Also it is possible to substitute hardware platforms with only little effort.

To save resources the CAN Multi Protocol Stack is configurable and scalable. The software tool CAN DeviceDesigner, which is part of the delivery, is used to do the configuration of the protocol stack.

This tool supports a fast and cost-saving design of devices. By predefined profiles and objects the tools helps to create consistent object dictionary and EDS files.



user interface CAN DeviceDesigner

<< CANopen Services <<

Services	Basic Slave	Master/Slave	Manager
SDO Server*	2	128	128
SDO Client*		128	128
SDO expedited/ segmented/block*	●/●/-	●/●/●	●/●/●
USDO Connections**	2	512	512
PDO Producer	32	512	512
PDO Consumer	32	512	512
PDO Mapping	static	static/dynamic	static/dynamic
MPDO Dest.Mode*		●	●
MPDO Src. Mode*		●	●
SYNC Producer		●	●
SYNC Consumer	●	●	●
Time Producer		●	●
Time Consumer		●	●
Emergency Producer	●	●	●
Emergency Consumer	1	127	127
Guarding Master*			●
Guarding Slave	●	●	●
Bootup Handling		●	●
Heartbeat Producer	●	●	●
Heartbeat Consumer		127	127
NMT Master		●	●
NMT Slave	●	●	●
Master Bootup CiA302-2*			●
Flying Master CiA302-2*		○	●
Configuration Manager CiA302-3*			●
SRD CiA302-5*		○	●
Redundancy CiA302-6*		○	○
CANopen Router CiA302-7*		○	○
LED CiA303	●	●	●
Safety*	○	○	○
LSS CiA305	●	●	●
Networking/Routing*	○	○	○
ISO-TP		●	●
Multiline		○	○
C#-API-Wrapper Windows	○	○	○

* Classic CAN Protocol Stack ** CAN FD Protocol Stack

● inclusive ○ optional

Services	J1939
node address assignment	static oder Adress Claiming
time monitoring of messages	●
cyclic transmission/reception	●
transport protocol (TP) for large data volumes	●

supported targets

The CAN Multi Protocol Stack of emotas may be used with or without operating system. Numerous CAN controller and CPU types are being supported. For example: Bosch, Infineon, Intel, NXP, ST Microelectronics, Texas Instruments, Atmel, Renesas, Microchip.

The list of supported hardware platforms is being continually extended:

<https://www.emotas.de/en/info/controller>

If your required target is not available, the development of further driver packages is possible without any problem.

scope of delivery

- ▶ CANopen/J1939/RawCAN Protocol Sourcecode (ANSI-C)
- ▶ Multiprotocol driver
- ▶ CAN DeviceDesigner
 - 1 Named User License (Project License Stack)
 - 1 Floating License with dongle (Site License Stack)
 - Network License via license server
- ▶ ready to run example application
- ▶ (reference) manual
- ▶ optional maintenance agreement
- ▶ optional integration support

licensing

- ▶ Project License (Royalty free; incl. 6 months phone and e-mail support)
- ▶ Site License without runtime fees (Royalty free; incl. 24 months phone and e-mail support)

order information

- ▶ 5031-10 | Multiprotocol J1939/CANopen Slave Stack
- ▶ 5004-11 | MP J1939/CANopen Master/Slave Stack
- ▶ 5004-12 | MP J1939/CANopen Manager Stack

All Multiprotocol stack variants are also available as FD, Multiline, or FD & Multiline

<< CANopen & J1939 trainings 2026 <<

Our on-site training courses primarily address development engineers and system integrators. We provide insights into the functionality and operation of the CAN based protocols CANopen and J1939.

For decision-makers, on the other hand, the training course offers an opportunity to familiarize themselves with the protocols and decide whether it is suitable for their applications.

Upcoming trainings at our site in Merseburg:

- 21st April: CAN (FD), ISO-TP and UDS
- 22nd April: CAN (FD) and J1939 (FD)
- 23rd April: CAN (FD) and CANopen (FD)

times: from 9 a.m. to 5 p.m.

costs: €370 per participant - early bookers receive a 15% discount up to 5 weeks before the date

- ▶ including training materials and certificate

Upcoming online trainings (4 hours):

- 29th April: CANopen
- 7th May: J1939

times: from 9:45 - 11:45 a.m. & 12:45 - 2:45 p.m.

costs: €190 per participant

- ▶ including digital training materials and certificate

Depending on your desired specialist topics and the respective level of knowledge of the participants, we will create individual in-house or remote training courses for you. Contact us - we're at your disposal.



Please note that the training courses are held in German.

<< CAN Hardware (selection) <<

CAN hardware interfaces enable PC applications to access CAN or CAN FD networks. Choose the optimal CAN interface for your application for CANopen (FD), J1939 or EnergyBus in accordance with performance conditions. For operation with the emotas software tools feel free to contact us – we will assist you.

Changing CAN hardware may be easily done, because most available CAN interfaces are appropriate for different system architectures. The manufacturers of the offered hardware products guarantee highest quality and extensive product support for your special application area.

SYS TEC electronic GmbH

<< USB-CANmodul <<

- High Speed CAN Interface
- 11 bit ID and 29 bit ID CAN support
- 9-pin D-SUB connector
- diagnostic LEDs
- CAN-Interface



EMS Wünsche

<< EtherCAN 3.0 <<

- Ethernet/CAN Gateway, 2xCAN, RS232, USB Host, SD Card Slot
- individual galvanic separation
- 454 MHz ARM9 CPU (Freescale I.MX287)
- Web Server
- 24 V supply
- platform for local control applications



Kvaser AB

<< Kvaser U100 <<

- USB-CAN/CAN FD interface, single-channel
- supports both 11-bit (CAN 2.0A) and 29-bit (CAN 2.0B active) identifiers.
- Galvanic isolation
- 9-pin D-SUB connector



Kvaser AB

<< Kvaser Leaf v3 <<

- supports CAN FD, up to 8 Mbit/s
- supports both 11-bit (CAN 2.0A) and 29-bit (CAN 2.0B active) identifiers.
- Galvanic isolation
- 9-pin D-SUB connector
- support for SocketCAN



<< J1939 Stack <<

With the emotas J1939 Protocol Stack users are able to develop J1939 devices quickly and easily. The Protocol Stack includes the full range of functions necessary for the integration of J1939 into own devices.

The comfortable API allows a simple evaluation of received data. The J1939 Stack has a defined driver interface which guarantees high flexibility, because of an easy adaption of the stack to existing or new CAN Controller or CPU types. Changing the hardware platform is therefore easy to realize. The usage of different Real Time Operating Systems (RTOS) or Linux (SocketCAN, can4linux), Windows and QNX is possible. The scope of delivery includes several examples and enables the users to have a simple and comprehensible introduction to the usage of the J1939 stack. Furthermore emotas provides a reference manual containing the description of function parameters.

<<configuration <<

In order to keep the consumption of resources as low as possible, the J1939 stack may be configured and scaled to a large extent. An individual adjustment to customer products can be realized by various options of configuration. The setting for these features is done by the graphical configuration tool J1939 DeviceDesigner. By the tool the user is able to configurate the PGNs and the mapping of SPNs with an integrated database.

<< properties <<

- transport protocols (BAM and CMTD) for large datasets
- setting of the node ID: fixed or via address claiming
- time monitoring of cyclical J1939 messages
- cyclical sending and receiving of J1939 messages
- address claiming protocol is completely supported
- dynamic configuration of PGNs and SPNs during runtime
- diagnostic messages (optional)
- ISO-TP client/server

Supported targets

The J1939 Protocol Stack of emotas may be used with or without operating system. Numerous CAN controller and CPU types are being supported. For example: Bosch, Infineon, Intel, NXP, ST Microelectronics, Texas Instruments, Atmel, Renesas, Microchip.

The list of supported hardware platforms is being continually extended:

<https://www.emotas.de/en/info/controller>

If your required target is not available, the development of further driver packages is possible without any problem.

scope of delivery

- ▶ Protocol Sourcecode (ANSI-C; MISRA-compliant)
- ▶ 1939 driver
- ▶ J1939 DeviceDesigner
 - 1 Named User License (Project License Stack)
 - 1 Floating License with dongle (Site License Stack)
 or: Network License via license server
- ▶ ready to run example application
- ▶ (reference) manual
- ▶ optional maintenance agreement
- ▶ optional integration support

licensing

- ▶ Project License
(Royalty free; incl. 6 months phone and e-mail support)
- ▶ Site License without runtime fees
(Royalty free; incl. 24 months phone and e-mail support)

order information

- ▶ 5028-10 | J1939 Stack Pack
- ▶ 5028-10-ML | J1939 Stack Pack MultiLine
- ▶ 2028-xx | J1939 driver
- ▶ 2028-xx-ML | J1939 driver MultiLine
- ▶ 0100-40 | integration support

<< J1939 FD Stack <<

The emotas J1939 FD stack contains the classic J1939 stack. Both at compile time and at runtime, you can choose between J1939 and J1939 FD.

The J1939 FD protocol stack enables the simple and fast development of J1939 FD devices according to J1939-22 with the existing functionality for the integration of J1939 FD into own devices. A comfortable API allows the simple evaluation of received data. The J1939 FD stack is equipped with a defined driver interface. The driver interface ensures a high degree of flexibility because the protocol stack can be easily adapted to existing or new CAN controllers or CPU types. This ensures an uncomplicated exchange of the hardware platform. The use of different real-time operating systems (RTOS) as well as Linux (SocketCAN, can4linux), Windows and QNX is possible. Several examples are included in the scope of delivery, so that it is easy to get started using the J1939 FD stack. The reference manual with a description of the function parameters is also available for using the functions.

<<configuration <<

In order to keep resource consumption low, the J1939 FD stack is largely configurable and scalable. Only the program parts required for the application are translated and integrated. The J1939 FD Library supports all relevant J1939 transport protocols and optionally diagnostic messages. Predefined SLOT functions also support the conversion of measured values into the transmission format and vice versa. Due to the complete implementation of the communication mechanisms defined by J1939 FD, developers can fully concentrate on the application.

<< properties <<

- transport protocols (BAM and CMTD) for large datasets
- fixed node address or address claiming
- time monitoring of cyclical J1939 messages
- cyclical sending and receiving of J1939 messages
- address claiming protocol is completely supported
- dynamic configuration of PGNs and SPNs during runtime
- optionally Diagnostic Messages extension with simplified API for DM1..DM64
- ISO-TP client/server
- MPG transport

Supported targets

The J1939 FD Protocol Stack of emotas may be used with or without operating system. Numerous CAN controller and CPU types are being supported. For example: Bosch, Infineon, Intel, NXP, ST Microelectronics, Texas Instruments, Atmel, Renesas, Microchip.

The list of supported hardware platforms is being continually extended:

<https://www.emotas.de/en/info/controller>

If your required target is not available, the development of further driver packages is possible without any problem.

scope of delivery

- ▶ Protocol Sourcecode (ANSI-C; MISRA-compliant)
- ▶ 1939 driver
- ▶ J1939 DeviceDesigner
 - 1 Named User License (Project License Stack)
 - 1 Floating License with dongle (Site License Stack)
 or: Network License via license server
- ▶ ready to run example application
- ▶ (reference) manual optional optional multi line support for gateways or multi-channel devices
- ▶ optional UDS server extension available
- ▶ optional maintenance agreement
- ▶ optional integration support

licensing

- ▶ Project License (Royalty free; incl. 6 months phone and e-mail support)
- ▶ Site License without runtime fees (Royalty free; incl. 24 months phone and e-mail support)

order information

- ▶ 5028-10-FD | J1939 FD Stack Pack
- ▶ 5028-10-FD-ML | J1939 FD Stack Pack MultiLine
- ▶ 2028-xx-FD | J1939 FD driver
- ▶ 2028-xx-FD-ML | J1939 FD driver Multiline

<< J1939 DeviceDesigner <<

The J1939 DeviceDesigner is an easy-to-use tool for fast and cost-saving design of J1939 devices. With a few mouse clicks users may merge PGNs and SPNs of a device from predefined databases and specifies the properties of signals.

Based upon this the J1939 DeviceDesigner generates the PGN and SPN configuration and initialising functions in C as well as the device documentation. Additionally the J1939 DeviceDesigner optimally configures the J1939 Stack and the driver package by means of device properties.

<< PGN and SPN configuration <<

According to the integrated database and inputs the PGN and SPN configuration is being generated as source code file in C, being integrated to the application as interface for the protocol stack. Depending on the configuration users may have access to the variables of the PGNs as C variable directly from the application.

<< stack configuration <<

Basing on the settings and definitions of the PGNs and SPNs a configuration and initialization file is being generated as source code file in C, ensuring that only actually used services of the J1939 stack are being compiled and initialized. Concerning the resources the storage needs of the J1939 stack is configured optimally.

<< device documentation <<

Related to the configuration of PGN and its settings and description as well as for further device specific settings a device documentation is generated in HTML and text format. This documentation is equal to the current settings and implementation and is therefore always up to date. You may furthermore use the generated documentation as a part of the user documentation of the device or import it into word processing programs.

system requirements

- ▶ Windows 11 or later
- ▶ diverse Linux variants (tested with (K)Ubuntu)
- ▶ emotas J1939 Protocol Stack

scope of delivery

- ▶ Software for Download
- ▶ PDF manual
- ▶ PGN/SPN databases

licensing

- ▶ Named User License (Project License)
- ▶ Floating License with dongle (Site License)
- ▶ Network License via license server
- ▶ Corporate License on request

order information

- ▶ 1030-10/11 | J1939 DeviceDesigner Windows/Linux
- ▶ 1030-10-D/11-D | J1939 DeviceDesigner Floating License Windows/Linux

Evaluation versions of the tool available upon request.

<< J1939 CANinterpreter <<

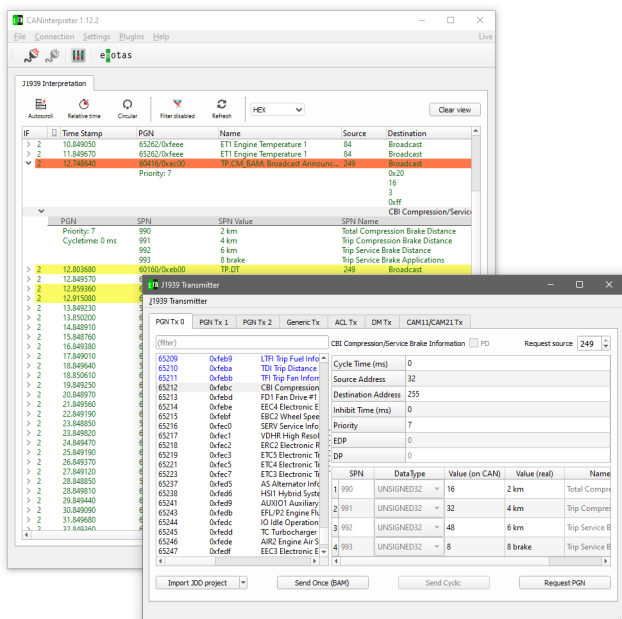
The CANinterpreter is a multi-purpose tool to monitor and record data traffic and for the analysis of CAN networks. By using the J1939 plugin messages are being interpreted and sent according to J1939.

<< basic features <<

- online-monitoring of bus traffic
- interpretation of CAN data according to J1939
- recording of CAN messages
- Flexible filtering/triggering of CAN-IDs
- single or cyclic sending of CAN messages/sequences

<< J1939 plugin <<

The J1939 interpretation interprets each CAN telegram according to J1939 standard. Each message is mapped with name as well as source and target address. The PGNs including the SPNs are being interpreted in detail with all containing values as physical variable of value and units. Besides the linear view of telegrams shown in the following figure a J1939 object view is also available, showing each signal with current values.



Besides this linear view of telegrams shown in the figure a J1939 object view is also available showing all signals including their current values. Interpretation may be completed by import options for für proprietary PGNs. BAM- und TP-message sequences are being analyzed completely.

The monitoring tool allows you to send and to request J1939 telegrams as well as to send address claiming telegrams. For software engineers and users of J1939 devices the CANinterpreter and its J1939 interpretation is an ideal add-on expandable by further optional plugins. Additional customer-specific plugins may be created on request. Example would be service or operation tools for plant and machinery.

<< scripting plugin <<

This plugin is based on QTScrip/JavaScript, extended by CAN and J1939 specific commands and may be used for the creation of own user interfaces, for example for testing and control applications. Design is done via the graphical UI designer. Ready to run examples are included in the scope of delivery.

<< J1939 transmitter plugin <<

This Plugin enables the sending of various J1939 messages:

- Standard PGNs <= 8 bytes
- Request PGN and Address Claiming
- Broadcast PGNs > 8 bytes by BAM
- some diagnostic messages
- J1939 CAM11/CAM21 messages

order information

- ▶ 5017-10/11 | CANinterpreter J1939 Windows/Linux
- ▶ 5017-10-D/11-D | CANinterpreter J1939 Floating Lizenz Windows/Linux

<< J1939 Bootloader (CAM11/CAM21) <<

An increasing number of J1939 devices need bootloaders to update firmware in the field. The use of a complete J1939 stack for the bootloader is mostly unsuitable, because of the large flash memory footprint. Emotas has developed a new J1939 Bootloader using CAM11/CAM21 that needs only few memory resources.

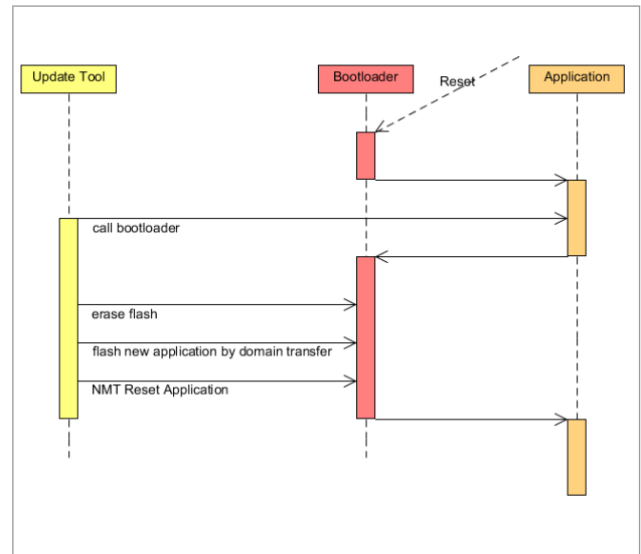
The bootloader supports only the most necessary services for a bootloader. The emotas bootloader therefore only requires 16 KiB of flash and is still J1939 compatible. The bootloader is available for various 16- and 32-bit microcontrollers and can be easily adapted to other target systems.

Manufacturer-specific program code can be easily integrated, since the bootloader is supplied in C source code. Therefore it is possible for device manufacturers to integrate e.g. access protection mechanisms, firmware encryption or a backdoor into the bootloader.

The application, which is to be transferred with the Bootloader, must be adapted slightly. To do this, the linker settings must be changed so that the reset vector points to the boot loader and the space is not taken up by the application. In addition, persistent RAM (4 bytes) must be provided for data exchange between the boot loader and the application, and alternative ways of communication between the application and the boot loader are also supported. The binary file of the application is also preceded by a data block which contains the CRC and other information. The corresponding space in the flash must also be kept free for this.

<< Highlights <<

- ▶ J1939 compatible
- ▶ segmented transfer and block transfer using CAM11/CAM21
- ▶ only little flash resources needed



system requirements

- ▶ ANSI-C Compiler

scope of delivery/licensing

- ▶ J1939 Bootloader source code (ANSI-C compatible)
- ▶ driver for CAN controller, CPU and flash access
- ▶ ready to run example application
- ▶ user manual and reference manual in electronic format
- ▶ incl. 6 months (Project License) or 24 months (Site License) support by e-mail or phone
- ▶ incl. 1 Named User License (Project), Floating License (Site) or Network License via license server for CAN UpdateManager

order information

- ▶ 5039-10 | J1939 (CAM11/CAM21) Bootloader generic

<< UDS Server/Client Stack Extension <<

Unified Diagnostic Services (UDS) is a communication protocol for the diagnosis of ECUs in the automotive environment, which is standardized in ISO 14229. Our UDS server/client stack extension implements UDS on CAN and is based on ISO-TP. ISO-TP is an international standard (ISO 15765-2) for the transmission of data via CAN. The protocol covers the OSI layers 3 and 4. Data packets with a length of up to 4095 bytes can be transmitted in segments.

<< UDS Server/Client Stack for UDSonCAN <<

The UDS server/client stack extension is available in C source code and has been developed according to the MISRA-C:2004 coding rules. Currently it covers the following UDS services:

- 10h DiagnosticSessionControl
- 11h ECUReset
- 14h ClearDiagnosticInformation
- 19h Read DTC Information
- 22h ReadDataByIdentifier
- 23h ReadMemoryByAddress
- 27h Security Access
- 28h CommunicationControl
- 2Ah ReadDataByPeriodicIdentifier
- 2Eh WriteDataByIdentifier
- 2Fh IOControlByIdentifier
- 3Dh WriteMemoryByAddress
- 31h RoutineControl
- 34h Request Download
- 35h Request Upload
- 36h Transfer Data
- 37h Request Transfer Exit
- 3eh TesterPresent
- 85h ControlDTCSetting

With both functional and physical ISO-TP addressing. The UDS server/client stack extension can be combined with the J1939 stack, the CANopen stack, the RawCAN stack and also with the CAN multi protocol stack and is available for a variety of targets - from 8 bit microcontrollers, over the arbitrary ARM cores, up to various PC platforms.

<<UDS Interpretation <<

UDS Interpretation is optionally available for our CANopen DeviceExplorer. Additional UDS commands may be sent by this UDS PlugIn.

system requirements

- ▶ CANopen Stack, J1939 Stack, RawCAN-Stack
CAN MultiProtokollstack
- ▶ ANSI-C Compiler

scope of delivery/licensing

- ▶ UDS Server sourcecode (ANSI-C compatible)
- ▶ example application
- ▶ PDF user and reference manual
- ▶ Licensing based on stack license form
- ▶ optional maintenance agreement
- ▶ optional integration support

order information

- ▶ 1033-10 | UDS Server Addon
- ▶ 1033-10-ML | UDS Server Addon Multiline
- ▶ 1033-11 | UDS Client Addon
- ▶ 1033-11-ML | UDS Client Addon Multiline

<< UDS Bootloader <<

A growing number of ECUs today require bootloaders to exchange firmware in the field. The use of a complete UDS stack is often unsuitable for bootloaders due to the large flash memory requirements. The UDS bootloader from emotas was therefore developed with a focus on low memory requirements, while at the same time taking into account high fail-safety.

The bootloader supports only the most necessary services for a bootloader. The emotas UDSONCAN bootloader therefore only needs 32 KiB Flash and is still UDS compatible. The bootloader is available for various 16- and 32-bit microcontrollers and can easily be adapted to other target systems.

Manufacturer-specific program code can be easily integrated, since the bootloader is supplied in C source code. This allows device manufacturers to integrate e.g. access protection mechanisms, firmware encryption or a backdoor into the bootloader.

The application, which is to be transferred with the Bootloader, must be adapted slightly. To do this, the linker settings must be changed so that the reset vector points to the boot loader and the space is not taken up by the application. In addition, persistent RAM (4 bytes) must be provided for data exchange between the boot loader and the application, and alternative ways of communication between the application and the boot loader are also supported. The binary file of the application is also preceded by a data block which contains the CRC and other information. The corresponding space in the flash must also be kept free for this.

<< UDS services and limitations <<

- 10h DiagnosticSessionControl
- 11h ECUReset
- 22h ReadDataByIdentifier
- 27h SecurityAccess
- 31h RoutineControl
- 34h RequestDownload
- 36h TransferData (download only)
- 37h RequestTransferExit

If further UDS services are required, we can extend the UDS bootloader on your behalf to include additional UDS services and specific "update sequences".

<< Highlights <<

- ▶ UDS compatible
- ▶ data transfer via ISO/TP
- ▶ only little flash resources needed

system requirements

- ▶ ANSI-C Compiler

scope of delivery/licensing

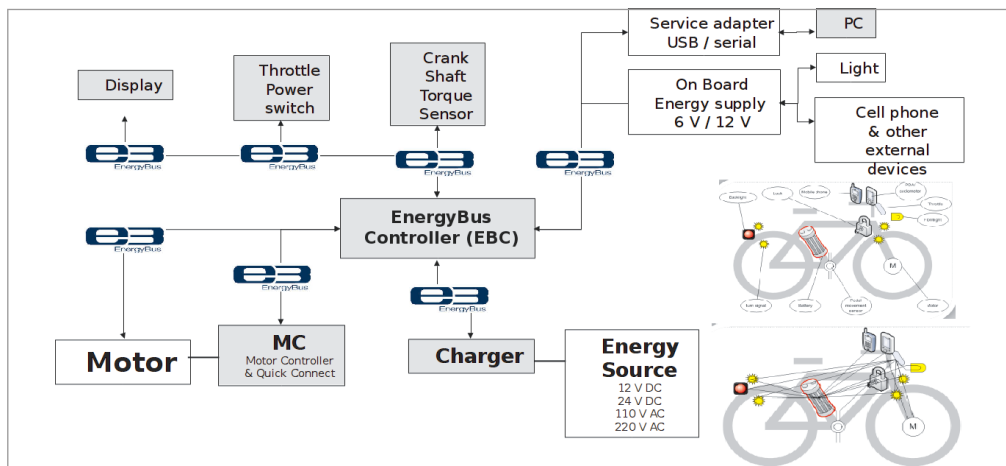
- ▶ UDS Bootloader Source Code (ANSI-C compatible)
- ▶ driver for CAN controller, CPU and flash access
- ▶ ready to run example application
- ▶ user manual and reference manual in PDF
- ▶ incl. 6 months (Project License) or 24 months (Site License) support by e-mail or phone
- ▶ incl. 1 Named User License (Project), Floating License (Site) or Network License via license server for CAN UpdateManager

order information

- ▶ 5041-10 | UDS Bootloader Pack

<< Introduction to EnergyBus <<

EnergyBus is an open standard for all electrical components of electric bicycles and LEVs (Light-Electric-Vehicles). Standardization ensures compatibility and safety of LEVs. EnergyBus consists of a specifically developed connector family and a communication protocol basing on CANopen.



<< EnergyBus communication <<

Concerning data exchange and communication EnergyBus bases on the proven CANopen protocol. Together with the CAN in Automation e.V. the members of the EnergyBus e.V. have been developing the specific characteristic of EnergyBus as a CANopen application profile and published as CiA 454. Each EnergyBus network requires an EnergyBus controller (EBC) as CANopen Master. Different components like battery, drive, display, charging cable lock etc. are defined as CANopen slaves with specific parameters and reactions. Besides transferred data and cycle times the application profile also includes mechanisms for detection of devices during the exchange of components and compatibility checks.

<< EnergyBus e.V. <<

The standardization of EnergyBus is being driven by the EnergyBus e.V. which composes different well-known companies from the fields of battery management, bike rental, energy supply, drive technology or components manufacturer for LEVs as well as dedicated private individuals. Founded in Taiwan in 2007 the association is nowadays situated in Thuringia.



In 2012 emotas became member and software development partner at the EnergyBus e.V. and is actively involved in further development of the EnergyBus communication protocol.

for more information:
www.energybus.org

<< CANopen requirements <<

Integrating EnergyBus into components a CANopen Stack or an EnergyBus Framework is required. Additionally, for the EnergyBus controller and external battery chargers CANopen Master functionalities are required. Furthermore, for all devices the support of LSS (Layer Setting Services, CiA 305) for automatic addressing is mandatory and the usage of a bootloader is being recommended. Because of the flexible PDO configuration with respect to runtime the CANopen implementation has to support dynamic object dictionary entries. The EnergyBus framework of emotas accomplishes all these requirements and furthermore contains besides the CANopen features EnergyBus specific functionalities

<< EnergyBus Framework <<

The EnergyBus LEV Framework provides an application programming interface for Light Electric Vehicle (LEV) devices according to the CiA 454 profile. Basing on the proven CANopen Stack, it enables the fast and easy integration of EnergyBus and the required CANopen services by a flexible API.

The modular structure of the EnergyBus LEV Framework allows an easy integration of different device types like display, drive, battery and EBC (EnergyBus Controller). Each single device can also be combined with the EBC.

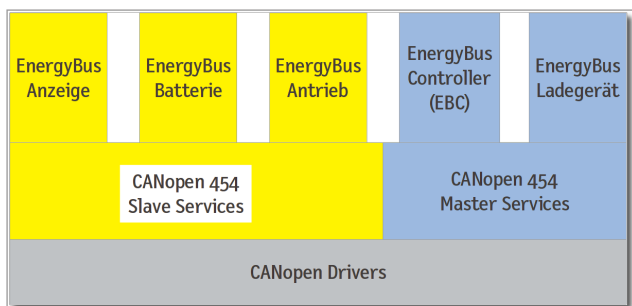
The data of the object dictionary depend on the device type. They are provided in a data block that can be read and written by the application. Information about changes of the objects are sent to the application by callback functions. The handling of the CANopen communication works automatically in the background.

Configurations are done with the EnergyBus DeviceWizard tool which creates the object directory, the related EDS files and a device documentation.

<< Slave Module <<

Currently following EnergyBus components are being supported by the EnergyBus Framework as Slave:

- display (HMI)
- battery
- drive (motor controller, MCU)
- Security Device
- Load Monitoring Device

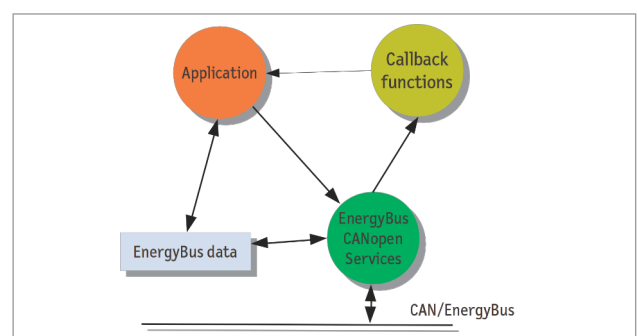


<< external battery charger <<

Containing both Master and Slave functionalities, external battery chargers are supported by a separate module of the framework. Depending on the connected components, the module performs only slave functionalities (with EBC in the system) or master functionalities in the network, i.e. LSS, node number assignment, NMT commands (without EBC in the system).

<< EnergyBus controller <<

The EnergyBus Controller (EBC) is the central control element of the EnergyBus network and does always function as master (LSS, NMT and master control). The EBC is also responsible for the assignment of node numbers, switching the NMT and EnergyBus state machine and for device control. You may chose whether you want to use the EBC as a separate device or together with other components in one device. It is commonly used in combination with drives (motor controller).



<< Services <<

Service	EB Slave Device	EB EBC	EB ext. charger
node number	via LSS	1	127
virtual devices	1	127	127
SDO Server	1	1	1
SDO Client		128	128
SDO expedited/ segmented/block*	●/●/○	●/●/○	●/●/○
PDO Producer	20	512	512
PDO Consumer	4	512	512
PDO Mapping	static	dynamic	dynamic
SYNC Producer		●	● ¹
SYNC Consumer	●		● ²
Emergency Producer	●	●	●
Emergency Consumer		127	127
Bootup Handling		●	●
Heartbeat Producer	●	●	●
Heartbeat Consumer	2	127	127
NMT Master		●	● ¹
NMT Slave	●		● ²
LSS CiA305/Fastscan Master	●	●	● ¹
LSS CiA305/Fastscan Slave Handling	●		

● inclusive ○ optional

1 - without EBC in network only

2- with EBC in network only

supported targets

The EnergyBus Framework may be used with or without operating system. Numerous CAN controller and CPU types are being supported.

The list of supported hardware platforms is being continually extended:

<https://www.emotas.de/en/info/controller>

If your required target is not available, the development of further driver packages is possible without any problem.

scope of delivery

- ▶ EnergyBus LEV Framework Source code (ANSI-C)
- ▶ ready to run example application
- ▶ (reference) manual
- ▶ optional maintenance agreement
- ▶ optional integration support

licensing

- ▶ Project License (Royalty free; incl. 6 months phone and e-mail support)
- ▶ Site License without runtime fees (Royalty free; incl. 24 months phone and e-mail support)
- ▶ incl. EnergyBus DeviceWizard
Project License: 1 Named User
Site License: 1 Floating License with dongle
or: Network License via license server

order information

- ▶ 1014-10 | EnergyBus LEV Framework Slave Communication Modul
- ▶ 1014-11 | EB LEV Framework Master/Slave Communication Modul
- ▶ 1014-20 | EnergyBus LEV Framework EBC module
- ▶ 1014-21 | EnergyBus LEV Framework Anzeige module
- ▶ 1014-22 | EnergyBus LEV Framework battery module
- ▶ 1014-23 | EnergyBus LEV Framework drive module
- ▶ 1014-24 | EnergyBus LEV Framework Voltage Converter Modul

<< EnergyBus DeviceWizard <<

The EnergyBus DeviceWizard is a software tool for user-friendly configuration, scaling and parameter setting of the EnergyBus LEV Framework. An easy to handle assistant leads the user step by step through the program and lets him choose between the EnergyBus devices (battery, charger, drive, display or EnergyBus-Controller(EBC)). During further steps the user defines the properties of the components. A deep understanding of communication protocols is not necessary.

The user is led through the configuration of the hardware (microcontroller) by the assistant of the EnergyBus DeviceWizard. Only some data about CPU type, CAN controller type and some timing parameters need to be chosen here (i.e. the frequency of microcontroller) and the driver will be configured according to the settings done before.

<< program output <<

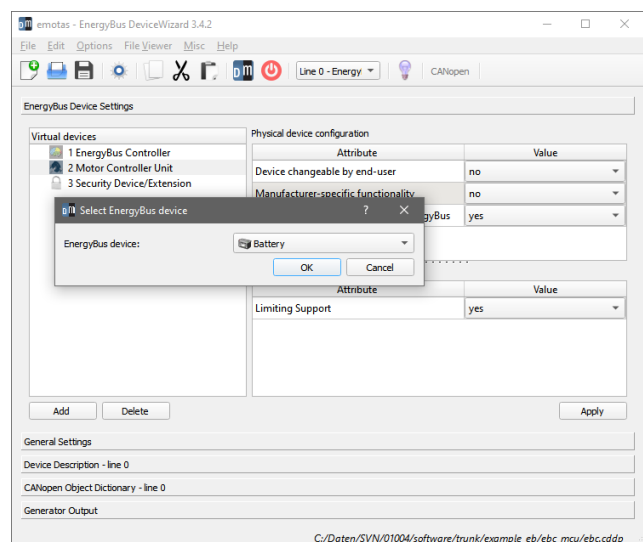
Basing on the entered data by the user the software generates .h configuration files, .c application code as well as required files for the technical manual in different formats. During the generation process the software verifies the plausibility of the given information and shows warnings if necessary. The configuration files configure the EnergyBus Framework and activate only particularly required functionalities of the framework by setting compiler switches.

optimal configuration of EnergyBus Framework with the EnergyBus DeviceWizard

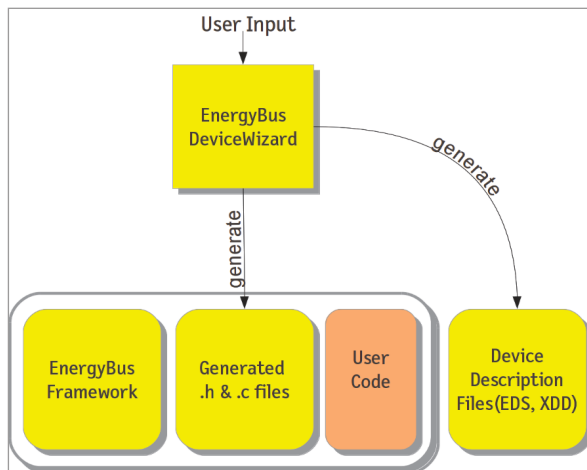
The precisely generated C files contain the CANopen object directory, the PDO mapping code and the application interface. The EnergyBus Framework is therefore especially designed to the application and optimized with regard to RAM and Flash.

<< expert modus <<

Furthermore, the expert modus allows CANopen-experienced users to create objects in the manufacturer-specific part of the object directory or device profile segment. CANopen objects may be also imported from CODB oder EDS files. If possible within the range of functions of the EnergyBus framework, the CANopen functionality of the generated application code may be extended.



<< Design Flow <<



<< generated files <<

- object dictionary
- EnergyBus Framework configuration
- EnergyBus Framework initialization
- Electronic Datasheet (EDS according to CiA 306)
- XDD according to CiA 1311
- object dictionary documentation in HTML
- object list in CSV format for export to spreadsheet programs
- object description in .txt format

system requirements

- ▶ Windows 11 or later
- ▶ diverse Linux variants (tested with (K)Ubuntu)
- ▶ emotas EnergyBus Framework

scope of delivery

- ▶ EnergyBus DeviceWizard Software for Download
- ▶ PDF manual

licensing

- ▶ Named User License
- ▶ Floating License with dongle
- ▶ Network License via license server
- ▶ Corporate license on request
- ▶ EnergyBus Framework incl.1 Floating Lizenz

order information

- ▶ 1013-10/11 | EnergyBus DeviceWizard Windows/Linux
- ▶ 1013-10-D/11-D | EnergyBus DeviceWizard Floating License Windows/Linux

Evaluation versions of the tool available upon request.

<< EnergyBus Tool <<

The EnergyBus Tool is used for the analysis and for monitoring purposes of EnergyBus components in LEVs. In the basic version all components, its current state and operating data are shown, but may be supplemented by plugins with various additional functions for monitoring EnergyBus components.

Current device states may be exported as XML files and may be later imported again. These files containing the current state of an EnergyBus network (i.e. pedelec) may be sent for service purposes as well. Furthermore the device status report may be issued as a pdf document.

<< optional extension <<

- EnergyBus message interpretation
- EnergyBus message filtering
- CAN-raw data export
- firmware update of components
- EBC simulation

Device Parameters (fixed)		Actual Values
Property	Value	
Vendor ID	10066329	
Product code	269746211	
Revision number	0.2	
Nominal voltage VDN 1	48.00 V	
Security properties	Mechanical locking with charge&lock cable supported Softlock is...	
Key class	External device; temporary keys required	

HMI control unit/display	
Device Parameters (fixed)	
Vendor ID	10066329
Product code	1
Device information	EBC controller supported Passive device Device not changeable by end-user Manufacturer-specific functionality supported Masterless mode supported Bootloader active; no application software is running
Nominal voltage VDN 1	48.00 V
Security properties	Mechanical locking with charge&lock cable supported Softlock is supported
Key class	External device; temporary keys required
Actual Values	
Device control command	No action
Device status	nothing active
EnergyBus FSA state	Compatibility Check
Device electronic temperature VDN 1	23.00 °C
Implemented specification versions	454.1.2.0 454.2.2.0 454.3.2.0 454.4.2.0 454.5.2.0 454.6.2.0 301.0.4.2 302.1.4.1 302.3.4.1 305.0.2.2 0.0.0.10
Security Status	No locking mechanism supported Device is not locked locally Manipulation detected Device owns current encryption key Device owns initial encryption key Locking of all virtual devices in the EMS requested All virtual devices in the EMS that support softlocking are locked

system requirements

- ▶ Windows 11 or later
- ▶ diverse Linux variants tested with (K)Ubuntu
- ▶ supported CAN interface

CAN interfaces

Windows®

- ▶ IXXAT
- ▶ SYS TEC USB-CANmodul
- ▶ PEAK PCAN-USB
- ▶ Kvaser CAN-Interfaces
- ▶ EMS Wünsche CAN-Interfaces
- ▶ Vector CAN-Interfaces (XL-Driver)

Linux®

- ▶ can4linux
- ▶ SocketCAN

scope of delivery

- ▶ EnergyBus Tool Software for download
- ▶ PDF manual

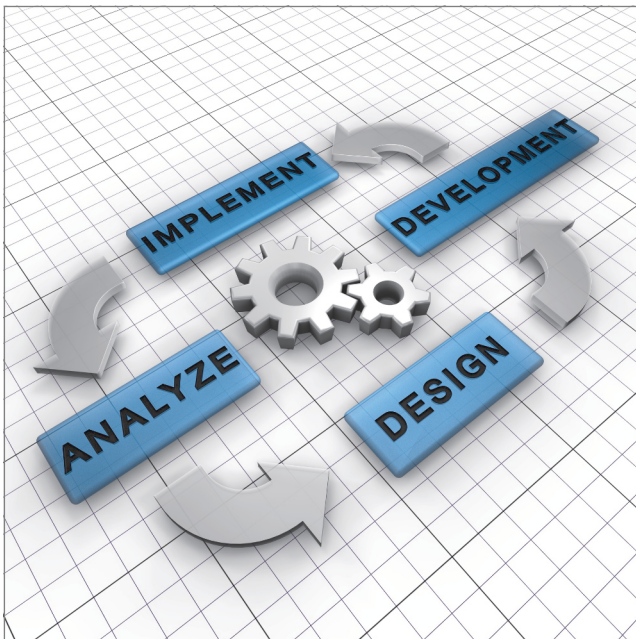
<< Embedded Software Development <<

Based on many years of experience and sound knowledge emotas focusses on software development for numerous microcontrollers and different customers' requirements: from 8 to 64 bit, from ARM to DSP, from singlecore to multicore. emotas supports customers during the whole process of device development.

All software engineers have long-standing experience in the field of software development for embedded devices used in medical technology, automation technology, renewable or conventional energy technologies and other sectors. For complex developments emotas provides a broad product range of software tools.

An increased complexity of software development is due to the requirement of a perfect interplay of hardware and software. All software engineers at emotas do have electrotechnical knowledge and expert knowledge in programming with C for microcontrollers and assemblers, in real-time operating systems and modern development tools.

With emotas, customers always have a personal contact person and a direct line to responsible engineers of their projects.



Get in touch with emotas for an initial discussion

From the first contact to a successful project conclusion and beyond that, emotas accompanies and supports you through all stages within the development process:

- joint preparation of specification sheet
- draft and design
- implementation
- test
- integration
- maintenance and updates

<< CANopen Integration <<

Are you developing a new device or do your customers desire a CANopen interface for an already existing device? emotas will support you by sound knowledge of CANopen and perform the integration of the CANopen Stack into your devices. Your requirements are the be-all and end-all.

emotas performs the communication links of your devices. Our software engineers have long-standing experience in the integration of Fieldbus stacks, especially of CANopen stacks into embedded devices used in medical technology, automation technology, renewable or conventional energy technologies and other sectors. emotas is familiar with the device-specific requirements and the device profiles of CANopen.

emotas is developing at the home location in Merseburg. The acquisition of the hardware will be done at emotas or at your location and may be combined with a CANopen training and an introduction to the CANopen Stack and tools.



emotas is always up-to-date, because we are active member of the CiA e.V. and we have been involved in different working groups for standardization. Together with your know-how we will find the optimal solution for your application.

<< projects implemented <<

- digital and analogue IO devices
- modular equipment families
- position and tilt sensors drives
- maritime sensors
- safety sensors/actuators
- medical devices
- SIIS level-2 sensors
- lift applications
- joysticks

Short development times by professional knowledge

From the first contact to a successful project conclusion and beyond that. emotas accompanies and supports you through all stages within the development process:

- choosing the appropriate device profile and objects
- joint preparation of specification sheet
- draft and design
- implementation
- integration of the CANopen Stacks
- preparation and implementation of the conformance tests
- adaption to bootloader
- maintenance and updates

your benefits with emotas:

- // **CANopen integration by experienced CANopen software engineers**
- // **fast time-to-market**
- // **calculable development costs**
- // **focusing on core competences**

<< EnergyBus Integration <<

Are you developing new components for LEVs? Or do you already provide one? You or your customers need an EnergyBus interface? Gain from our know-how. We'll support you and perform the integration of the required EnergyBus software into your devices.

emotas performs the communication links of your devices. Our software engineers have long-standing experience in the integration of CANopen Stacks and EnergyBus Stacks in LEV components. We're familiar with the device-specific requirements set by the EnergyBus standard.

emotas is developing at the home location in Merseburg. The acquisition of the hardware will be done at emotas or at your location and may be combined with a training or an introduction to the EnergyBus Framework and tools.



emotas is also active member of the EnergyBus e.V. and has been involved in different working groups, too. Together with your know-how we will find the optimal solution for your application.

.....
Competence for your needs
.....

From the first contact to a successful project conclusion and beyond that, emotas accompanies and supports you through all stages within the development process:

- choosing the appropriate device profile and objects
- joint preparation of specification sheet
- draft and design
- implementation
- integration of the EnergyBus Framework
- preparation and implementation of the conformance tests
- adaption to bootloader
- maintenance and updates

<< project examples <<

- battery and charger
- motor controller
- EnergyBus controller
- HMI/Display

your benefits with emotas:

- // **EnergyBus integration by experienced EnergyBus software engineers**
- // **fast time-to-market**
- // **calculable development costs**
- // **focusing on core competences**

<< Commissioning Support <<

emotas ensures the communication (CAN/EtherCAT) of the protocol stacks in your devices. By a professional commissioning support emotas helps you to integrate the communication stacks into your devices promptly.

Everything you'll need is your hardware, your development environment, a software engineer of your company and the long-standing experience of emotas. During the joint commissioning support one of the numerous emotas example projects for common evaluation boards is used and ensures an easy transfer of the software to customers' hardware.

Little 'obstacles' like the setting of frequency, register offset, transceiver configurations or something similar may be overcome together. Furthermore you'll gain from valuable tips on the used protocol, the internal mode of action and on the interfaces of the emotas protocol stacks.

Subsequently you are able to link the communication stack with your application.

Furthermore we would like to recommend an additional training to ensure a fast and successful start of the project.



your benefits with emotas:

- // **adaption and integration of the stacks by emotas' software engineers**
- // **fast time-to-market**
- // **calculable development costs**
- // **focusing on core competences**

<< Trainings <<

People starting with a new topic are advised to use our trainings and seminars for CAN/CANopen, J1939 and EnergyBus. By individual agreements of the topics emotas addresses both, beginners and pros.

<< target group <<

Our seminars addresses to software engineers who want to become familiar with the technologies, establish a basis for further work with emotas products or want to know more about specific device and application profiles. Furthermore we also address decision-makers who want to become more familiar with the technical opportunities of the technologies mentioned before.

.....
Contact us - we're at your disposal
.....

<< contents <<

Our trainings are individually tailored to your requirements and your (prior) knowledge. The following contents should be therefore seen as examples:

- introduction to CAN
- introduction to CANopen (FD)/EnergyBus/J1939
- explanations about emotas' stacks and available extensions
- overview of device and application profiles
- detailed informationen about specific device or application profiles
- commissioning support bases on application examples

Based on the knowledge of the participants we'll create an individual seminar schedule for different topics.

<< our trainers <<

Based on a long-standing experience our trainers are capable of imparting knowledge and has become respected experts and co-trainers at seminars of the CAN in Automation e.V. (CiA). During their active membership in the CiA and by a steady work for the standardization of CANopen and EnergyBus emotas' trainers are always up-to-date.

<< training material <<

A comprehensive training material will be available for any participant.

<< location and time of trainings <<

We offer trainings as in-house seminars in our premises at Merseburg. Depending on the content the seminars are structured for one or two days. Combined with an integration support you may be well prepared for a fast and easy start of a new project. Appointments will be done in consultation with our participants.



<< 10 Good reasons to decide for emotas <<

Nowadays there are numerous software companies in the automation industry. Choosing an appropriate partner being able to implement individual requirements, making the decision anything but easy. We would like to give you 10 reasons to decide for the products of emotas.

1. The team of emotas possesses competence and long-standing experience in the field of CAN/CANopen.
2. As an active member emotas has been involved in the standardization in the CiA. Furthermore emotas is member of the Technical Committee of the CiA e.V. and therefore always up-to-date.
3. the emotas CANopen/CANopen FD and J1939 source code is tested on MISRA conformity.
4. emotas' products are being improved steadily and being adapted to current standards.
5. Customers and users gain from practice-oriented trainings for stacks and tools and are able to start their projects promptly.
6. emotas provides comfortable graphic configuration and installation tools included in the scope of delivery of the stacks as well as an comprehensive tool chain for CANopen, EnergyBus, J1939 and UDS.
7. Customers gain from a fast and free First-Level-Support with direct line to emotas' software engineers.
8. emotas has been development partner of the EnergyBus association and is well networked with further partners. This allows us to give you the best possible advice.
9. User-friendly license terms grant the use of the source code for different projects. Customers may choose between a site or Project License.
10. Our customers stay always up-to-date by individual maintenance agreements at a fair price and gain from long-term support and important software updates.



<< Partners (selection) <<



<< About emotas <<

Being situated in Central Germany (Saxony-Anhalt) emotas embedded communication GmbH was founded by the software engineers Andreas Boebel, Torsten Gedenk and Steffen Rose as a company for software development. Together with a team of qualified software developers emotas is able to provide innovative and quality products for industrial communication and its customers.

emotas – your embedded solution partner

Thematic focuses of the product portfolio are CAN, CANopen, CANopen FD, EnergyBus, J1939 and UDS. Besides this emotas provides tailor-made embedded software solutions for customers. During this process customers can rely on a professional consulting from the first contact to the development, the commissioning support and the maintenance agreement. emotas is always focussing on the customers' needs. Additionally emotas offers training about the communication protocols CANopen, EnergyBus and UDS as well as individually agreed topics.

emotas is member of the CAN in Automation e.V. and development partner of the EnergyBus e.V. By an active cooperation in working groups and committees for standardization at the CiA, EnergyBus e.V., PNO and ETG the software engineers of emotas are always up-to-date. Therefore emotas' products always comply with the latest technical standards.

A well working cooperative partnership with our customers is an important concern. We have understood that a successful future of emotas can only be realized by a close cooperation with our customers. They gain from maximum benefit by professional service, sound consulting and quality products.



<< contact <<

fon +49 3461 79416-0
fax +49 3461 79416-10
service@emotas.de
www.emotas.de

emotas embedded communication GmbH
Fritz-Haber-Str. 9
D-06217 Merseburg
Germany

<< Exclusion of Liability <<

All delivered programs and documentations of emotas have been compiled with great care, have been checked and tested in different scenarios. Nevertheless we assume no guarantee or liability for the software consisting no errors and being suitable for a particular purpose. Technical data and descriptions do not represent warranted properties in the legal sense. Consequential damages appearing due to the usage of the software are excluded from every liability or juristic responsibility. All rights of the programs and the documentation remain with emotas embedded communication GmbH. The copying or the forwarding in whole or in part to third parties require the express prior written approval of emotas embedded communication GmbH. The terms and conditions are part and parcel of any quotation of emotas and can be requested at any time. Brand names and products names not named explicitly are trademarks of the respective companies. Technical changes, mistakes and misprints reserved.

copyright

2025 emotas embedded communication GmbH
 Fritz-Haber-Str. 9
 D-06217 Merseburg
 Germany

fon +49 3461 79416-0
 fax +49 3461 79416-10
 service@emotas.de
 www.emotas.de

<< picture credits <<


- 3d small people – repairers © Anatoly Maslennikov – Fotolia.com – 3d small people – team © Anatoly Maslennikov – Fotolia.com – Computer learning © kablitzsch – Fotolia.com – Four main steps for a software process cycle © Yabresse – Fotolia.com - 3d small people - info icon © Anatoly Maslennikov - Fotolia.com - figures CAN Hardware © from the specified manufacturer - AURA award © emotas embedded communication GmbH

<< trademarks <<

All used trademarks and brands are the property of their respective owners. CiA®, CANopen® and CANopen FD® are registered Community Trademarks of CAN in Automation e.V.

UDS PGN Master
Slavetack ISO-TP Tools
SPN Data Transmission
Address Claiming Cont
roller Area Network CA
NUDS Data Link Layer C
AN Standardization Tr
ansport Protocol PGN
Messages CANopen RAW Car

contact

 **emotas embedded communication GmbH**
Fritz-Haber-Str. 9
D-06217 Merseburg
Germany

mail service@emotas.de
web www.emotas.de
fon +49 3461 79416-0
fax +49 3461 79416-10 ■

